

# REGULADOR DE CÀRREGA SOLAR AMB ALGORISME DE CÀRREGA INTERACTIU



Grau en Enginyeria Informàtica – Enginyeria de Computadors

Departament d'ESAI – Facultat d'Informàtica de Barcelona (FIB)

Universitat Politècnica de Catalunya (UPC) – BarcelonaTech

**Marc Hernández Folguera**

Director: Joan Climent Vilaró

Data defensa: 29/04/2016

## Resum

L'ús de les energies renovables per aconseguir electricitat, és una tendència que cada dia s'està fent més popular. Això és degut a la creixent preocupació per una més que probable desaparició dels combustibles fòssils. Malauradament hi ha un lent avanç en la tecnologia quan es tracta de les energies renovables, principalment per la seva falta d'eficiència i simplicitat. Per això, amb la intenció de fomentar l'ús de l'energia solar, reduint així la dependència amb aquests combustibles, s'ha desenvolupat un regulador de càrrega solar. Aquest regulador optimitza tant la vida útil de les bateries, com el seu temps de càrrega. Utilitza un algorisme de càrrega interactiu, el qual proporciona el millor corrent possible a aquestes. Una altra característica que aporta aquest dispositiu és la disponibilitat de configurar les funcions de seguretat i monitorització per part de l'usuari.

## Resumen

El uso de las energías renovables para conseguir electricidad, es una tendencia que se está haciendo cada día más popular. Esto es debido a la creciente preocupación por una más que probable desaparición de los combustibles fósiles. Desgraciadamente hay un lento avance en la tecnología cuando se trata de las energías renovables, principalmente por su falta de eficiencia y simplicidad. Por eso, con la intención de fomentar el uso de la energía solar, reduciendo así la dependencia de estos combustibles, se ha desarrollado un regulador de carga solar. Este regulador optimiza tanto la vida útil de las baterías, como su tiempo de carga. Utiliza un algoritmo de carga interactivo, el cual proporciona el mejor corriente posible a estas. Otra característica que aporta este dispositivo es la disponibilidad de configurar las funciones de seguridad y monitorización por parte del usuario.

## Abstract

Obtaining energy from renewable sources is growing ever popular, due to the growing concerns that one day fossil fuel energy sources will run out. Sadly there is a slow uptake of renewable energy technologies due to lack of efficiency and simplicity in systems. Therefore with the intention to encourage the use of solar power, reducing fossil fuel dependency, a solar charge controller has been developed. This controller optimizes both the batteries' lifetime and its charging time. It uses an interactive charge algorithm, which provides the best possible current to the batteries. Another feature of this device is that it allows the user to configure the security and monitoring functions of the system.

# Índex

## Contingut

Figures i Taules.....	6
1. Introducció.....	7
1.1. Contextualització .....	7
1.2. Formulació del problema .....	8
1.3. Estat de l'art .....	8
1.4. Solució proposada.....	9
1.5. Actors implicats.....	10
1.5.1. Desenvolupador .....	10
1.5.2. Tutor del projecte .....	10
1.5.3. Clients.....	10
1.6. Objectiu.....	11
1.6.1. Objectius Materials .....	11
1.6.2. Objectiu Acadèmic .....	12
1.7. Abast .....	12
1.8. Obstacles.....	13
2. Metodologia i rigor .....	14
2.1. Anàlisi de l'entorn .....	14
2.2. Requisits del maquinari.....	14
2.3. Simulació del sistema.....	14
2.4. Sistema físic.....	14
2.5. Anàlisi del sistema.....	15
2.6. Mètodes .....	15
2.6.1. Desenvolupament de cicles curts .....	15
2.6.2. Feedback del client .....	15
2.7. Consideracions.....	15
3. Descripció de les tasques.....	16
3.1. Planificació del projecte i viabilitat .....	16
3.2. Anàlisi i disseny .....	16
3.3. Fase de desenvolupament .....	16
3.4. Etapa final .....	17
3.5. Relació de precedència .....	17

3.6. Càlculs de temps .....	18
3.6.1. Temps estimat.....	18
3.6.2. Temps real.....	18
3.6.3. Diagrama de Gantt.....	19
3.7. Recursos.....	21
4. Maquinari .....	22
4.1. Concepte bàsic de funcionament .....	22
4.2. Interfície de càrrega.....	23
4.2.1. PWM.....	23
4.2.2. Estratègies de commutació.....	25
4.2.3. Divisor de tensió.....	27
4.3. Tria dels components.....	27
4.3.1. Microcontrolador.....	27
4.3.2. Sensor de corrent.....	29
4.3.3. Sensor de temperatura .....	30
4.3.4. Transistors.....	30
4.4. Disseny inicial.....	31
4.5. Disseny final .....	32
5. Algorisme .....	34
5.1. Optimització de càrrega .....	34
5.2. Protecció i monitorització .....	37
5.2.1. Protecció .....	37
5.2.2. Monitorització.....	38
6. Desenvolupament.....	39
6.1. Entorn de desenvolupament .....	40
6.1.1. Desenvolupament hardware .....	40
6.1.2. Desenvolupament software.....	41
6.2. Parts del programari .....	41
6.2.1. Biblioteques de perifèrics .....	41
6.2.2. Programari .....	44
7. Gestió econòmica .....	49
7.1. Identificació de costos .....	49
7.1.1. Recursos materials .....	49
7.1.2. Costos directes per activitat .....	49
7.1.3. Costos indirectes.....	50

---

7.1.4.	Amortitzacions .....	50
7.2.	Estimació de costos.....	51
7.3.	Control de gestió.....	51
7.3.1.	Desviaments.....	51
7.4.	Cost unitari.....	52
8.	Sostenibilitat .....	53
8.1.	Dimensió econòmica.....	53
8.2.	Dimensió social .....	53
8.3.	Dimensió ambiental .....	54
8.4.	Puntuació .....	54
9.	Conclusions.....	56
9.1.	Anàlisi d'objectius .....	56
9.1.1.	Objectius Materials .....	56
9.1.2.	Objectiu Acadèmic .....	57
9.2.	Conclusió personal .....	57
9.3.	Línies de futur .....	58
10.	Glossari .....	59
11.	Bibliografia i recursos web .....	61
12.	Annex A: Esquemes elèctrics .....	63
13.	Annex B: Codis font .....	64

## Figures i Taules

### Figures

Figura 1. Solució proposada.....	10
Figura 2. Diagrama de Gantt de la planificació inicial, part gràfica.....	19
Figura 3. Diagrama de Gantt final, part gràfica.....	19
Figura 4. Diagrama de Gantt part textual.....	20
Figura 5. Imatge representativa del sistema.....	23
Figura 6. Exemples funcionament d'un PWM.....	24
Figura 7. Exemples commutació per tall de la massa.....	25
Figura 8. Exemples commutació per tall del positiu.....	26
Figura 9. Divisor de tensió.....	27
Figura 10. Llei d'Ohm.....	30
Figura 11. Diagrama de blocs inicial.....	31
Figura 12. Diagrama de blocs final.....	32
Figura 13. Algorisme de càrrega estàndard d'una bateria.....	34
Figura 14. Algorisme de càrrega de 5 etapes.....	36
Figura 15. Font d'alimentació.....	40
Figura 16. Oscil·loscopi.....	40
Figura 17. Multímetre digital.....	41
Figura 18. Diagrama de flux general.....	45
Figura 19. Diagrama de flux inicialització.....	46
Figura 20. Fórmules pel càlcul dels desviaments.....	52
Figura 21. Diagrama Arduino Uno.....	63
Figura 22. Diagrama sensor de temperatura DHT11.....	63
Figura 23. Diagrama sensor de corrent.....	63
Figura 24. Diagrama relé de desconnexió.....	63
Figura 25. Diagrama transistor N-Mosfet.....	63

### Taules

Taula 1. Dedicació estimada per tasca, en hores.....	18
Taula 2. Dedicació real per tasca, en hores.....	18
Taula 3. Especificacions Arduino Uno.....	29
Taula 4. Utilització de recursos per activitat del Gantt.....	49
Taula 5. Costos i amortitzacions dels recursos materials.....	50
Taula 6. Costos dels recursos humans.....	50
Taula 7. Estimació inicial del cost total del projecte.....	51
Taula 8. Cost total del projecte.....	51
Taula 9. Cost unitari del regulador de càrrega solar.....	52
Taula 10. Matriu de sostenibilitat del TFG.....	54

## 1. Introducció

### 1.1. Contextualització

Un dels problemes més greus avui en dia és la cerca d'energies renovables que competeixin tant en prestacions com en rendibilitat amb els combustibles fòssils utilitzats, ja que és un recurs energètic cada cop més escàs.

Segons l'estudi fet per l'equip dirigit per Fabian Wagner<sup>[1]</sup>, l'energia solar és capaç de servir un percentatge substancial de l'energia elèctrica necessària per satisfer la demanda mundial.

Per tant explotar els beneficis que pugui aportar aquesta energia és una aposta amb bones perspectives de futur. I la clau rau en el sistema d'emmagatzematge.

Actualment hi ha dos tipus d'energia solar:

- El sistema d'**energia solar fotovoltaica**, que utilitza panells solars que converteixen la llum solar directament en electricitat<sup>[2]</sup>.
- Les plantes d'**energia solar concentrada**, que utilitzen l'energia tèrmica solar per fer vapor que per mitjà d'una turbina es converteix en electricitat<sup>[3]</sup>.

L'energia solar fotovoltaica és extremadament dinàmica tant en ús com en tecnologia. L'any 2013 un total de 142 GW d'energia solar van ser transformats en energia elèctrica: un 97,6% d'aquesta electricitat va ser produïda per plaques fotovoltaïques, mentre que el 2,4% restant va venir de l'energia solar concentrada<sup>[4]</sup>. A finals de l'any 2014, la tecnologia fotovoltaica va avançar, arribant a recollir un mínim de 178 GW, suficient per subministrar un 1% del total de l'electricitat utilitzada en tot el món<sup>[5]</sup>.

Espanya és un dels països més avançats en el desenvolupament d'energia solar, ja que és un dels països d'Europa amb més hores de sol. Actualment n'és el quart fabricant mundial<sup>[6]</sup>.

Un dels problemes principals de l'energia solar però, és que no es pot utilitzar més enllà de les hores diürnes, fent que el seu proveïment sigui limitat. Per tal de poder fer-la servir de forma ininterrompuda s'ha de recol·lectar energia extra durant les hores de llum i emmagatzemar-la de manera que es pugui utilitzar durant la nit si fos necessari.

Les bateries per emmagatzemar aquesta energia, quan es parla d'energia solar fotovoltaica, són les anomenades **bateries de cicle profund** (o *Deep cycle batteries* en anglès)<sup>[7]</sup>. Estan dissenyades per poder descarregar quasi tota la seva càrrega sense patir danys, podent aprofitar així tota la seva capacitat. Utilitzar entre un 45% i un 75% (depenent del fabricant i la construcció) de la càrrega és l'opció òptima per aquest tipus de bateries, encara que es pot arribar a descarregar fins a un 80% si és necessari<sup>[8]</sup>.

Aquest treball s'ha centrat en desenvolupar un algorisme interactiu per a la càrrega d'aquestes bateries amb la màxima eficiència possible, pels motius següents :

- Les energies renovables preocupen molt la societat moderna.
- L'energia solar és un sector molt important dins les renovables.
- Espanya és un dels països més desenvolupats en aquest sector degut en part a la gran quantitat d'hores de sol disponible en comparació a d'altres països d'Europa.

### 1.2. Formulació del problema

El Sol és una de les principals fonts d'energia d'aquest planeta, però no sempre és fàcil d'aprofitar-lo per a l'ús personal. El fet que la radiació solar es pugui convertir en electricitat és de gran utilitat, ja que pràcticament tota la tecnologia utilitzada actualment funciona amb electricitat.

Tot i disposar de tecnologies que ens permeten fer aquesta conversió, el seu rendiment no és suficient i no es pot deixar de dependre d'altres energies predominants, com són el petroli, el carbó o el gas natural.

El desenvolupament d'un algorisme de càrrega interactiu per un regulador de càrrega solar és un projecte que vol aconseguir un millor aprofitament de l'energia solar fotovoltaica.

La intenció d'aquest treball és poder donar eines als ciutadans per tal que aconseguixin una major eficiència energètica quan utilitzin energies renovables, la qual cosa pugui fomentar-ne el seu ús i, al mateix temps, reduir l'ús d'energies fòssils, ja que aquestes són finites ( i algunes contaminants).

El problema plantejat consisteix en analitzar un algorisme de càrrega de bateria de cicle profund a través de plaques solars i actuar-hi de tal manera que la càrrega sigui optimitzada. L'objectiu ha estat crear un regulador de càrrega solar per tal que l'energia de la bateria s'obtingui més ràpid i al mateix temps no se'n perdi quan no s'utilitza, ja que les bateries es poden anar descarregant fins i tot quan no estan en ús.

La creació d'un sistema capaç d'actuar d'aquesta manera requereix una sèrie de sensors, com per exemple sensors de corrent o de temperatura, i altres components hardware, com ara un microcontrolador, una pantalla LCD, transistors o un productor de senyals acústiques.

### 1.3. Estat de l'art

A la Gerència de Control i Instrumentació (GCI) de l'Institut d'Investigacions Elèctriques (IIE) es va desenvolupar un sistema de monitoratge remot i control de càrrega de bateries en aplicacions fotovoltaïques [9] [10] [11] que està basat en un microcontrolador que atorga la facilitat d'implementar, en un software, diferents tècniques per estimar l'estat de càrrega de la bateria. En la versió actual, l'estat de càrrega de les bateries es determina utilitzant una tècnica de recompte d'ampers/hora.



La universitat d'Stanford també va fer un estudi sobre bateries<sup>[12]</sup>, tot i que aquest estudi estava basat específicament en la programació d'un algorisme lineal i dinàmic de càrrega i descàrrega de bateries per cotxes elèctrics a través del qual podien allargar la vida de les bateries elèctriques fent-les així més eficients i més duradores.

L'any 2010, el departament d'Enginyeria Mecànica de la Universitat Nacional de Taiwan va realitzar una investigació sobre el dany que causa a les bateries la seva sobrecàrrega, i va descobrir, també, que degut a les mesures que es posen per evitar-la, les bateries redueixen normalment en gran mesura la seva capacitat de càrrega màxima. Acabaven proposant un nou algorisme de càrrega en el qual la capacitat augmentava substancialment, i a més prevenia el problema de la sobrecàrrega<sup>[13]</sup>.

Si bé és veritat que aquests projectes no són exactament el que es proposa en aquest treball, la gran similitud que hi ha entre aquestes investigacions i aquest estudi és prou important per què s'hagin tingut en compte a l'hora de desenvolupar-lo, ja que tots tracten una part de l'eficiència de les bateries i la millora del seu funcionament.

En el 2004, es va fer una tesi en el *Centro Nacional de Investigación y Desarrollo Tecnológico*<sup>[14]</sup>, sobre la determinació de l'estat de càrrega de bateries mitjançant xarxes neuronals. En aquesta tesi, es presentava el disseny, simulació, implementació i proves realitzades a una xarxa neuronal implementada en un controlador de càrrega basat en un microcontrolador. Si bé és cert que aquest projecte tampoc és exactament el mateix que el que aquí es presenta, tenen grans similituds. Aquí es pretén, sobretot, optimitzar la càrrega de les bateries, i no s'utilitzaran xarxes neuronals en cap moment del procés.

#### 1.4. Solució proposada

Tots els altres estudis mencionats prèviament es basen en, o bé allargar la vida de les bateries, o en determinar el seu estat de càrrega. Aquest projecte, a més a més de tenir tots aquests àmbits en compte, es diferencia de la resta perquè aporta un nou mètode interactiu de càrrega de bateries mitjançant el qual un regulador de càrrega varia el corrent que reben les bateries des de les plaques solars per tal de reduir així el temps necessari de càrrega. Això també evita possibles costos addicionals, com s'ha mencionat anteriorment.

Actualment existeixen algorismes que modifiquen tant la tensió com el corrent de càrrega de les bateries, però com es menciona en paràgrafs anteriors, la gran majoria estan destinats a allargar la vida de les bateries (per exemple, evitant sobreescalfaments). La solució que es proposa aquí és un algorisme que es comença de zero, tot i que és possible que s'hagin aprofitat conceptes o coneixements d'altres opcions ja creades.

Degut a l'ànima del projecte (sostenibilitat energètica), els elements utilitzats disposen d'un maquinari de baix consum així com un programari optimitzat pel seu ús.

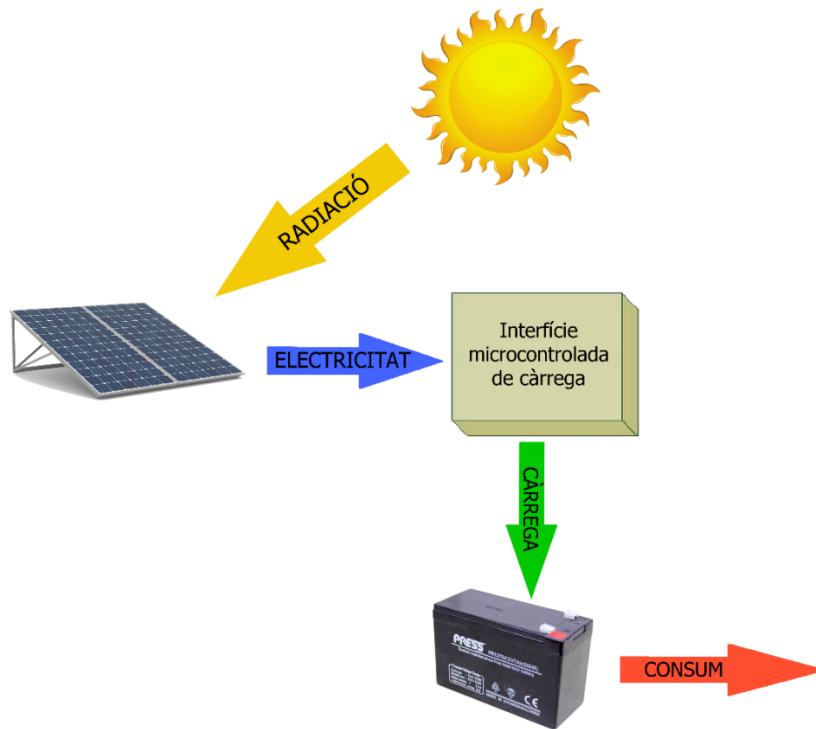


Figura 1. Solució proposada

## 1.5. Actors implicats

### 1.5.1. Desenvolupador

L'encarregat de desenvolupar el projecte ha estat el propi estudiant. Ha estat l'encarregat d'executar totes les fases del projecte, des de la planificació i l'estudi dels àmbits tractats fins al disseny i implementació tant del hardware com del software necessaris pel seu funcionament.

### 1.5.2. Tutor del projecte

El tutor del projecte ha estat qui ha supervisat, com a enginyer, el desenvolupament de totes les fases i qui ha anat guiant en el rumb correcte. Aconsellant en la presa de decisions en les fases de disseny i controlant el compliment del calendari i dels objectius marcats.

### 1.5.3. Clients

Aquest projecte, donat que el més important és el sol, pot tenir abast a nivell mundial, si bé és cert que per certs països podria ser més útil (Espanya, Austràlia o la Índia) que per d'altres (Regne Unit o Irlanda). Un algorisme que optimitzi la càrrega de bateries pot tenir una repercussió important en les economies d'una gran part de països de tot el món.

Els avanços aconseguits en el treball poden millorar la vida tant d'individus singulars, com podria ser gent que visquin en refugis, masies, aldees, cases a la muntanya, etc.. on fer-hi arribar electricitat pot ser difícil, fins a governs, que podrien reduir costos en la producció d'electricitat, amb la conseqüent millora de l'economia, i permetent-los amb l'estalvi aconseguir millorar i invertir en altres àmbits d'igual importància.

### 1.6. Objectiu

El projecte consisteix en el disseny i implementació d'un **regulador de càrrega solar digital**, el qual és l'encarregat de controlar constantment l'estat de càrrega de la bateria i regular la seva intensitat, preservant la vida útil de la mateixa a partir dels valors de tensió i corrent que rep del panell solar fotovoltaic. El regulador garanteix que el corrent de càrrega mai sigui excessiu per tal d'evitar gasificacions que, a banda de la seva toxicitat, malmeten la vida útil de la bateria.

Actualment els reguladors de càrrega introdueixen microcontroladors per a la correcta gestió del sistema fotovoltaic, de manera que permeten un control capaç d'adaptar-se a les diferents situacions de forma automàtica, així com la modificació manual dels seus paràmetres de funcionament en instal·lacions especials. També memoritzen dades que permeten conèixer quina ha estat l'evolució de la instal·lació durant un temps determinat. Per aquest fi consideren els valors de tensió, temperatura, intensitat de càrrega i descàrrega, i la capacitat de la bateria.

S'ha desenvolupat el maquinari i el programari necessari per aquest sistema amb les tècniques de disseny dels sistemes encastats. S'ha pretès també, experimentar amb un nou algorisme de càrrega que optimitzés el temps de càrrega de la bateria sense malmetre la seva vida útil.

#### 1.6.1. Objectius Materials

El regulador de càrrega solar digital té implementades les següents funcions:

- Funcions orientades a optimitzar la càrrega de la bateria:
  - Regulació del corrent de càrrega.
  - Programació del corrent màxim de càrrega.
  - Subministrament del corrent de manteniment a plena càrrega (corrent de flotació).
  - Impedir la descàrrega de la bateria sobre els panells.
- Funcions orientades a la protecció de la bateria:
  - Protecció per sobretensió en els panells solars.
  - Protecció contra sobre-consum de corrent de càrrega.
  - Desconnexió de consum per baixa tensió de la bateria.
  - Desconnexió per sobreescalfament de la bateria.

- Funcions orientades a la monitorització del sistema:
  - Avís per baixa tensió de la bateria.
  - Avís de desconexió de la bateria.
  - Avís per sobreescalfament de la bateria.
  - Visualització de la tensió de la bateria.
  - Visualització de la tensió del panell solar.
  - Visualització de la fase de càrrega (% càrrega).

Fent així un sistema programable **de càrrega, protecció i monitorització** de la bateria d'un sistema d'acumulació d'energia fotovoltaica el més complet possible.

#### 1.6.2. Objectiu Acadèmic

La càrrega de les bateries es pot portar a terme per dos mètodes: a corrent constant o a tensió constant. L'inconvenient del primer mètode és que s'ha de dissenyar un generador de corrent constant, i això implica l'ús d'electrònica analògica, amb els costos addicionals de disseny, fiabilitat i circuit imprès. Per altra banda, la càrrega a tensió constant, té l'inconvenient que a mesura que la bateria es va carregant la seva tensió s'aproxima a la tensió constant de càrrega. Això té per efecte disminuir progressivament el corrent de càrrega que absorbeix la bateria, la qual cosa provoca un **increment del temps total de càrrega**.

Per tal de resoldre aquests problemes s'ha desenvolupat un algorisme interactiu que proporciona el corrent de càrrega òptim en tot moment. El mètode consisteix en mesurar el corrent de càrrega a partir d'un **sensor de corrent** i buscar el valor òptim per aproximacions successives ampliant la diferència de potencial entre la tensió de càrrega i la tensió de la bateria. Amb aquest mètode la bateria rep sempre la tensió que genera el corrent òptim.

#### 1.7. Abast

Per poder escollir correctament el hardware adient per al sistema primer s'ha hagut d'analitzar quin seria el ventall de funcionalitats que es voldrien desenvolupar. Com s'ha esmentat prèviament, les funcionalitats que s'havien previst implementar havien estat les d'optimització de càrrega, les de protecció de la bateria i les de monitorització del sistema. Totes les proves d'aquest projecte s'han realitzat en el laboratori d'ESAI ja que allà disposen dels materials electrònics necessaris per simular l'efecte dels panells solars, i per tant s'evita tenir una dependència del temps atmosfèric.

Inicialment va caldre estudiar cada tipus de funcionalitat per determinar els dispositius externs necessaris. Depenent del nombre de funcions que es volien implementar en cada un dels apartats mencionats prèviament, els dispositius havien de ser més o menys potents, per la qual cosa el preu també variaria. Més funcionalitats implicaria major recollida de dades i espai de memòria, de la mateixa manera, si el nombre de funcionalitats hagués estat més reduït implicaria menys esforç tant energètic com econòmic.

Un cop determinat l'espai on realitzar les proves i els tipus de dispositius externs que s'havien d'utilitzar, es va procedir al seu desenvolupament. Per fer-ho, ha estat imprescindible disposar d'un maquinari preparat.

També s'ha realitzat el desenvolupament de l'algorisme calculant els corrents i voltatges òptims per cada moment de la bateria.

La primera prova del sistema real s'ha realitzat amb els dispositius configurats seguint el disseny representat en el diagrama de blocs que es mostra més endavant. En la primera prova, es va utilitzar un únic dispositiu, el qual segons les dades rebudes se li han hagut d'anar ajustant paràmetres com el temps de captació de les dades per tal d'obtenir la millor relació entre la informació obtinguda i la vida de les bateries.

### 1.8. Obstacles

Un primer obstacle rau en el propi algorisme de càrrega de les bateries. Donat que actualment aquesta tecnologia està molt avançada, trobar-ne un que sigui innovador i eficient, i/o que encara no hagi estat utilitzat a aquestes alçades, ha estat un dels punts més crítics i importants del projecte.

Un altre problema inicial ha estat l'excés de dades captades i passades al microcontrolador que no era capaç d'emmagatzemar totes les necessàries i comportava un greu problema, ja que feia que el patró de càrrega prengués decisions errònies.

Tot i no ser un problema, cal mencionar el canvi de tutor durant el transcurs del projecte ja que la dinàmica de treball es va veure lleugerament afectada. S'havia previst que el projecte es presentaria abans de la jubilació de Pere Marès Martí, però degut als contratemps explicats anteriorment referits a la recepció dels materials, no va ser possible. És per això que es va canviar el tutor, i va passar a ser-ne Joan Climent Vilaró. Cal destacar la rapidesa i eficiència del nou tutor.

Un altre fet conegut que va suposar un cert inconvenient és l'horari limitat dels laboratoris de la Universitat on es va treballar en el desenvolupament del regulador de càrrega ja que no eren compatibles amb l'horari de treball de l'estudiant.

Per últim, el fet que alguns dels dispositius del laboratori estiguessin avariats o tinguessin un funcionament erroni va suposar una notable pèrdua de temps. Per exemple, fonts d'alimentació que no donaven suficient voltatge o multímetres amb fusibles fosos. Alguns d'aquestes problemes es detectaven ràpid o eren de ràpida solució però d'altres requerien una investigació del desenvolupament del regulador abans d'atribuir la responsabilitat a les eines del lloc de treball.

## 2. Metodologia i rigor

### 2.1. Anàlisi de l'entorn

Per la creació de l'estructura formada pels sensors, microcontrolador, interfícies i fonts d'alimentació, es va haver de disposar d'un laboratori que permetés interactuar amb ells segons les diferents necessitats: connexió dels sensors amb el microcontrolador, programar el microcontrolador i al mateix temps tingués les eines necessàries pel control del voltatge, intensitat, etc... La sala utilitzada ha estat el laboratori del departament d'ESAI.

El següent pas va ser analitzar els espais de que es disposaven per a la instal·lació del producte un cop l'estructura estigués acabada. Va ser necessari determinar quines zones permetrien obtenir més dades rellevants per al sistema. Per un costat, interessava una àrea en la que la llum solar hi fos en abundància i el sol es veiés en tot moment del seu recorregut (des de que surt fins que es pon), com per exemple un camp. Per un altre costat però, també es va considerar interessant tenir en compte zones on el sol només hi fos de forma puntual però on el dispositiu desenvolupat seguís sent necessari, per exemple en una casa a la muntanya on l'electricitat fos difícil d'obtenir. Aquests casos més complicats són un dels reptes que aquest projecte també pretén solucionar.

### 2.2. Requisits del maquinari

Els components necessaris han estat la bateria, el microcontrolador, els dispositius externs (com per exemple sensors) i altres dispositius electrònics (com ara resistències òhmiques). Es va determinar primer quines eren les especificacions necessàries fent un estudi de les E/S tant digitals com analògiques, les interfícies, els canals de comunicació, els perifèrics a utilitzar, la capacitat de memòria del programa, juntament amb el nombre de fonts d'alimentació i/o les bateries.

### 2.3. Simulació del sistema

Encara sense els materials, es va començar a desenvolupar l'algorisme d'una manera "virtual". Sense l'estructura no es podia acabar de comprovar el seu funcionament al 100%, però sí que es podien començar a fer càlculs matemàtics per esbrinar per exemple el corrent òptim de la bateria.

### 2.4. Sistema físic

Un cop es va disposar del maquinari bàsic i del nou algorisme de càrrega per la bateria, es va procedir al muntatge de tota l'estructura (connexió dels sensors amb el microcontrolador, la bateria i el panell solar, entre d'altres).

Connectar els sensors al microcontrolador va significar afegir-li el software necessari perquè entengués les dades enviades pels sensors, al mateix temps que es triaven les entrades i sortides correctes que els havien d'enllaçar.

La comunicació entre la bateria i/o el panell solar sobre el seu estat es va fer a través del microcontrolador i dels sensors mateixos, mostrant els percentatges i voltatges en una pantalla LCD situada a la mateixa estructura.

## 2.5. Anàlisi del sistema

Finalment, va caldre comprovar el funcionament de tota l'estructura de manera conjunta. Amb totes les unitats instal·lades als seus llocs corresponents, es va verificar des de la pantalla LCD i des d'un ordinador que totes les dades s'enviessin correctament, que l'energia rebuda es llegís de forma adequada, al mateix temps que l'algorisme prengués l'acció adequada i fes que la bateria es carregués més ràpid.

També es va comprovar que cap component de l'estructura patís cap problema (com per exemple el sobreescalfament), així com que el software funcionés correctament (que no hi hagués cap desbordament de memòria, cap accés "null" a memòria, etc....). L'ús d'instruments alternatius, com ara el multímetre va ser de gran utilitat, sobretot per les comprovacions de lectures de voltatge i corrent.

Les dades preses han estat estudiades, comparades entre elles i mesurades amb aparells alternatius, per demostrar que en diferents condicions el sistema seguia funcionant a la perfecció.

## 2.6. Mètodes

Tenint en compte el període de temps de que es disposava per realitzar el projecte, era lògic pensar que una metodologia àgil era l'òptima per poder complir amb el temps estipulat. No obstant, les metodologies àgils més conegudes, com per exemple SCRUM o XP (eXtreme Programming) estan orientades a treballs en grup, i per tant seguir-les completament no va ser una opció vàlida. Tot i que, certs conceptes d'aquesta metodologia sí que són útils per projectes individuals.

### 2.6.1. Desenvolupament de cicles curts

Una metodologia de desenvolupament de cicles curts és la que s'ajustava millor a les necessitats del treball. Es tracta de proposar objectius en cicles curts, en aquest cas, setmanals. D'aquesta manera el seguiment és més precís, assegura que es segueix la planificació i es té una visió real de l'estat del projecte. A més a més, en el cas de trobar errors, se sap ràpidament d'on provenen minimitzant així riscos.

### 2.6.2. Feedback del client

Per altra banda, és important tenir *feedback* del client. Com que en aquest projecte no hi ha un client real, el tutor ha assumit aquest rol, de manera que ha comprovat l'avanç del projecte amb la màxima freqüència possible. S'han produït reunions setmanals per tal de definir nous objectius i fer aquestes comprovacions.

## 2.7. Consideracions

Considerant la naturalesa del projecte es va contemplar la possibilitat de desviacions o modificacions respecte la planificació inicial. Aquestes podien venir donades per l'aparició de nous requisits durant l'evolució del projecte o dificultats presentades a l'hora del seu desenvolupament. Com s'ha estipulat en el mètode a seguir, s'han utilitzat tècniques àgils, les quals donaven peu a introduir canvis en l'estratègia a seguir.

### 3. Descripció de les tasques

#### 3.1. Planificació del projecte i viabilitat

La primera tasca realitzada en el projecte va ser la planificació i l'estudi de la viabilitat. Es va determinar quin era l'abast a partir del problema formulat. A continuació es va procedir a l'extracció de totes les tasques que eren necessàries per la realització del projecte, així com establir una metodologia de treball que garantís el compliment dels objectius plantejats.

Un cop determinat el pla a seguir i les tasques a realitzar, es va establir el pressupost que es considerava necessari pel projecte tenint en compte les hores i recursos que es preveien utilitzar, així com els seus costos associats.

L'últim pas en la planificació va ser conèixer l'estat de l'art i així disposar d'un punt de partida més sòlid que validés i orientés la decisió inicial sobre la viabilitat del projecte. Per això va ser necessari conèixer quines solucions similars hi ha en el mercat i veure que s'hi podia aportar.

#### 3.2. Anàlisi i disseny

Aquesta etapa requeria un estudi dels diferents àmbits que es tractarien durant el projecte així com el disseny del pla d'execució.

Principalment s'ha comprovat que els requisits estipulats es seguissin en tot moment, ja que aquests permetien assolir els objectius marcats, base de l'èxit del projecte. Cada un dels objectius havia de complir-se amb els requeriments que s'hi establien específicament.

Establerts els requisits necessaris, el següent pas ha estat dissenyar un pla d'execució de les tasques a desenvolupar en la implementació. Aquest disseny inclou tant els sistemes *hardware* com el *software*. Per tant, es va iniciar amb una anàlisi de tot el maquinari que s'utilitzaria i es va continuar amb el programari que havia de disposar el sistema. Cal destacar que l'ús del *hardware* específic i de baix consum crea per ell mateix unes limitacions de recursos que el disseny del *software* ha de tenir en compte indispensablement.

Cal afegir, a més, que un cop obtingudes les funcionalitats requerides pel sistema, s'ha hagut d'establir un pla de verificació de totes així com un sistema de proves per determinar els ajustaments necessaris.

Per tant, el disseny del projecte s'ha basat en aquests tres punts: *hardware*, *software* i *sistemes de verificació*.

#### 3.3. Fase de desenvolupament

La fase de desenvolupament del projecte es divideix en els següents apartats:

- Escollir els dispositius externs (senyors de corrent, LCD, sensor de temperatura, polsadors, senyal acústic, commutadors, ..... ) més adients pel nostre disseny.
- Dissenyar les interfícies d'adaptació necessàries per poder connectar els dispositius externs.
- Estimar la capacitat necessària de memòria del programa (Flash).



- Generar un diagrama de blocs del sistema on es mostri:
  - I. El microcontrolador genèric amb les interconnexions als diferents sensors, actuadors i blocs funcionals externs (detallant els busos amb el nombre de línies i la seva direcció amb coherència amb els apartats anteriors). Interfícies i les seves connexions.
  - II. Les fonts d'alimentació i/o bateries amb els voltatges corresponents.
- Estudiar l'entorn de treball.
- Estudiar i dissenyar el sistema físic. En base al material adquirit, estudiar el seu funcionament a nivell tècnic. Cada fabricant pot precisar dels seus propis programes informàtics específics per al seu funcionament (compilador, instal·lador, etc...). Per tant, ha estat necessari aprendre a utilitzar-los i configurar-los. També ha estat indispensable conèixer en profunditat les característiques del *hardware*, del que s'han hagut d'estudiar els manuals dels sensors.
- Implementar el *hardware* i el *software*. Es tracta del disseny i muntatge de la infraestructura, des de les connexions entre sensors i controlador fins el disseny i programació de tot el codi a implementar.
- Verificar el sistema i fer ajustaments. L'última fase del desenvolupament va consistir en la creació de les proves per a la validació del funcionament de tota la infraestructura i, segons els seu resultats, la seva configuració.

### 3.4. Etapa final

Amb tot el sistema implementat i disposant de la infraestructura real en funcionament, només quedava fer una anàlisi del sistema i la corresponent extracció de conclusions. En aquesta etapa s'ha recollit la documentació i s'ha escrit l'informe final, així com la preparació de la presentació corresponent.

### 3.5. Relació de precedència

Tant les tasques de **Planificació del projecte i viabilitat** com l'**Anàlisi i disseny del projecte** es van començar de forma simultània doncs no depenien de cap altra tasca.

La fase de **Desenvolupament** també es va poder començar al mateix temps, ja que la feina inicial no necessitava cap treball previ. En un cert punt però, sí que es va haver de prioritzar la part de l'anàlisi i disseny, perquè els seus resultats eren necessaris pel seguiment del desenvolupament. Tanmateix, l'**etapa Final** que depèn de tots els apartats previs, consisteix en el redactat de la memòria del projecte i per tant, la finalització completa del treball.

### 3.6. Càlculs de temps

#### 3.6.1. Temps estimat

A l'inici del treball es va estimar que les hores de dedicació del projecte serien les següents:

Tasca	Dedicació (hores)
Planificació del projecte i viabilitat	60
Anàlisi i disseny	60
Desenvolupament	280
Etapla final	50
<b>Total</b>	<b>450</b>

Taula 1. Dedicació estimada per tasca, en hores

#### 3.6.2. Temps real

Ha resultat però, degut en gran mesura als obstacles trobats, que les hores necessàries per poder finalitzar el projecte han estat aquestes:

Tasca	Dedicació (hores)
Planificació del projecte i viabilitat	60
Anàlisi i disseny	160
Desenvolupament	380
Etapla final	50
<b>Total</b>	<b>650</b>

Taula 2. Dedicació real per tasca, en hores

### 3.6.3. Diagrama de Gantt

El diagrama de Gantt dissenyat en la planificació inicial es mostra a continuació:

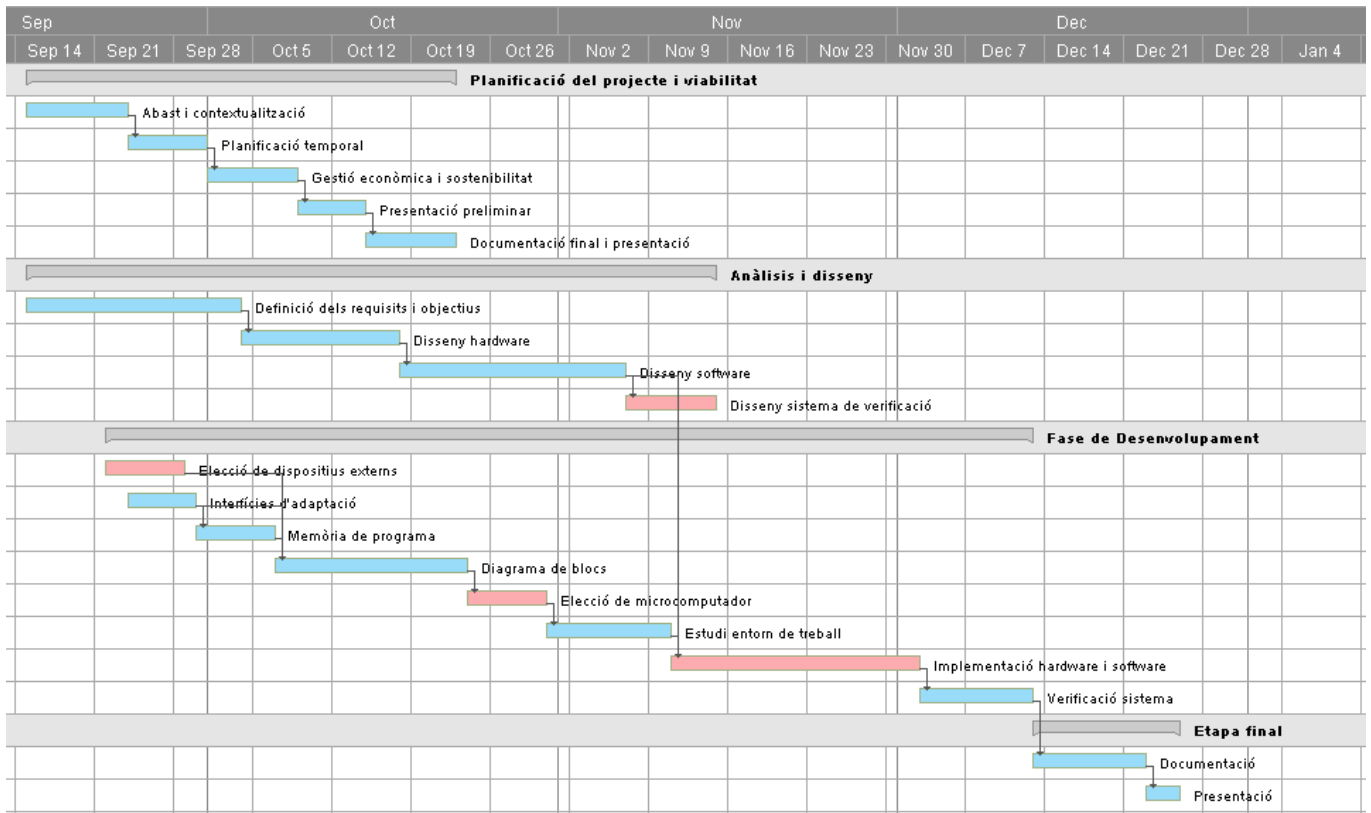


Figura 2. Diagrama de Gantt de la planificació inicial, part gràfica

Tot i que el projecte ha acabat seguint el següent diagrama:

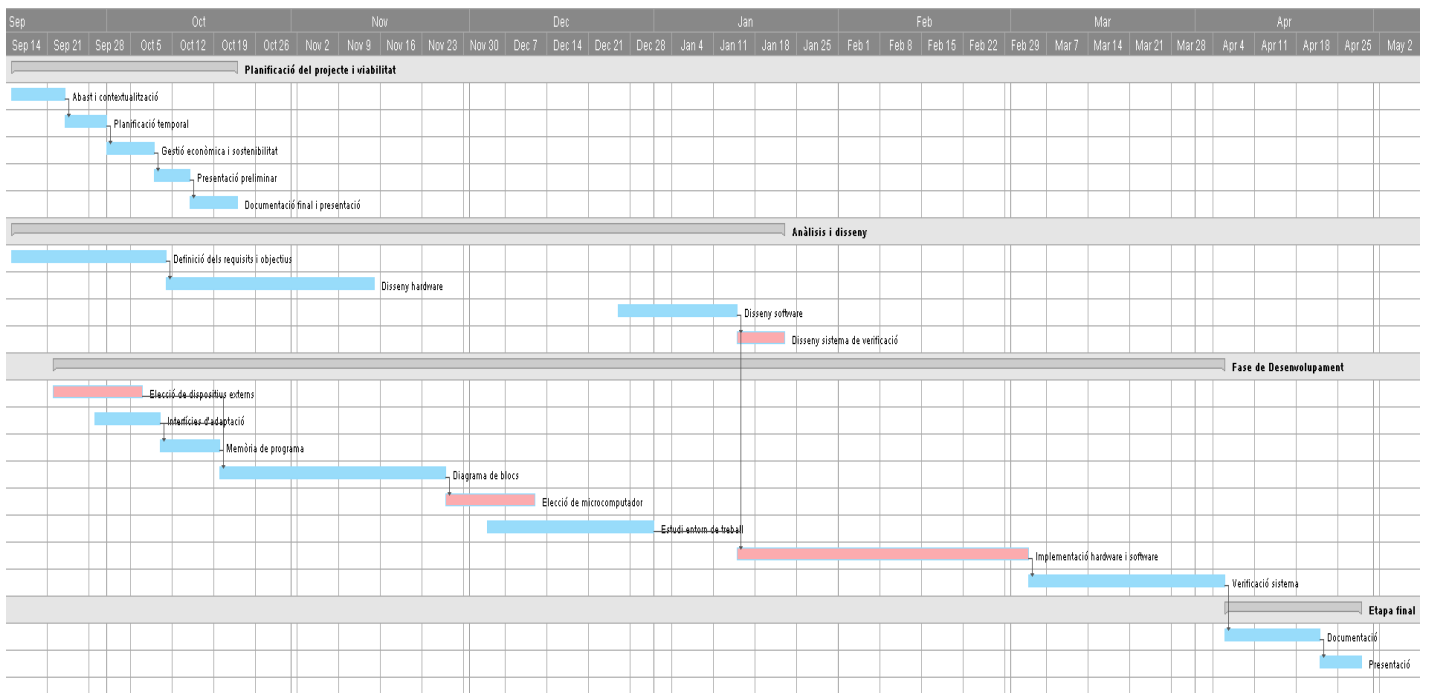


Figura 3. Diagrama de Gantt final, part gràfica

	Task Name	Start Date	End Date	Assigned To
1	<b>Planificació del projecte i viabilitat</b>	<b>15/09/15</b>	<b>22/10/15</b>	
2	Abast i contextualització	15/09/15	23/09/15	Becari d'investigació
3	Planificació temporal	24/09/15	30/09/15	Cap de projecte
4	Gestió econòmica i sostenibilitat	01/10/15	08/10/15	Becari d'investigació
5	Presentació preliminar	09/10/15	14/10/15	Cap de projecte
6	Documentació final i presentació	15/10/15	22/10/15	Becari d'investigació
7	<b>Anàlisis i disseny</b>	<b>15/09/15</b>	<b>22/01/16</b>	
8	Definició dels requisits i objectius	15/09/15	10/10/15	Enginyer de suport a la investigació
9	Disseny hardware	11/10/15	14/11/15	Enginyer de suport a la investigació
10	Disseny software	26/12/15	14/01/16	Enginyer de suport a la investigació
11	Disseny sistema de verificació	15/01/16	22/01/16	Enginyer de suport a la investigació
12	<b>Fase de Desenvolupament</b>	<b>22/09/15</b>	<b>05/04/16</b>	
13	Elecció de dispositius externs	22/09/15	06/10/15	Becari d'investigació
14	Interfícies d'adaptació	29/09/15	09/10/15	Becari d'investigació
15	Memòria de programa	10/10/15	19/10/15	Becari d'investigació
16	Diagrama de blocs	20/10/15	26/11/15	Enginyer de suport a la investigació
17	Elecció de microcomputador	27/11/15	11/12/15	Becari d'investigació
18	Estudi entorn de treball	04/12/15	31/12/15	Enginyer de suport a la investigació
19	Implementació hardware i software	15/01/16	03/03/16	Enginyer de suport a la investigació
20	Verificació sistema	04/03/16	05/04/16	Becari d'investigació
21	<b>Etapla final</b>	<b>06/04/16</b>	<b>28/04/16</b>	
22	Documentació	06/04/16	21/04/16	Becari d'investigació
23	Presentació	22/04/16	28/04/16	Cap de projecte

Figura 4. Diagrama de Gantt part textual

### 3.7. Recursos

Els recursos utilitzats en el transcurs del projecte es poden classificar en tres apartats:

- *Humans*. El projecte ha estat desenvolupat per l'estudiant de forma individual. Aquest ha rebut el suport i la supervisió dels dos directors al llarg del treball, Pere Marès Martí i Joan Climent Vilaró, ambdós professors del departament d'ESAI.
- *Hardware*. Els recursos *hardware* estan formats per un ordinador, que ha permès la programació de tot el *software* necessari, una font d'alimentació, un multímetre, un oscil·loscopi, un soldador, un microcontrolador, una bateria, una placa de connexió "*Breadboard*", així com un seguit de sensors i components electrònics, com resistències òhmiques, díodes, etc... Les característiques concretes d'aquests components es troben en els annexos. Alguns dels quals han estat cedits pel mateix departament d'ESAI.
- *Software*. Les eines utilitzades pel projecte, deixant de banda la del desenvolupament són les de gestió i documentació del projecte. Per la implementació de l'algorisme s'ha fet servir el programari *Arduino IDE*, pels diagrames de Gantt s'ha utilitzat la pàgina web *Smartsheet*, i per la redacció de tots els documents necessaris s'ha fet servir el *Microsoft Office 2013*. El desenvolupament complet d'aquest projecte ha estat fet amb el sistema operatiu *Windows 10*. A més, s'ha utilitzat tant *Google Drive* com *Dropbox* per facilitar l'accés dels documents i arxius des de diferents punts així com a còpia de seguretat d'aquests.

Més endavant es pot trobar una descripció de tots els recursos utilitzats en el projecte i, d'una forma més detallada, els de l'etapa de desenvolupament.

## 4. Maquinari

### 4.1. Concepte bàsic de funcionament

Tal i com es refereix a la introducció, cal desenvolupar el maquinari necessari per permetre una interconnexió controlada entre la bateria i el panell solar, ja que si es connectés directament seria impossible complir la majoria d'objectius, tant acadèmics com materials. Cap algorisme interactiu podria ser aplicat i per tant no es podria optimitzar la càrrega de la bateria, ni es podria garantir la seva màxima durada. A més a més, podria arribar el punt que la bateria es descarregués, donant potència al panell solar (fent anar el corrent en direcció contrària), en hores de foscor per exemple. Per arribar al nivell de funcionament desitjat, la connexió necessària entre el panell solar i la bateria serà gestionada per un microcontrolador.

El mètode de càrrega que seguirà aquest projecte és per corrent continu, però evitant haver de fer un generador de corrent constant i, per efecte directe, l'ús d'electrònica analògica.

La manera de gestionar la càrrega de la bateria serà utilitzant la tècnica de PWM (*pulse-width modulation*) per controlar el corrent que arribi a la bateria. Aquesta tècnica serà administrada per un Arduino el qual, a través de les lectures enviades per un sensor de corrent informarà de l'amperatge rebut a la bateria i modificarà el pols del PWM per ajustar-lo a les especificacions desitjades.

És important entendre que el microcontrolador Arduino sempre s'alimentarà de la bateria garantint així l'autonomia del sistema.

El sistema permetrà:

- La connexió microcontrolada entre la bateria i el panell.
- Informar l'usuari de l'estat del sistema.
- Adaptar-se als canvis d'intensitat lumínica.

La Figura 5 és una imatge representativa de tot el sistema.

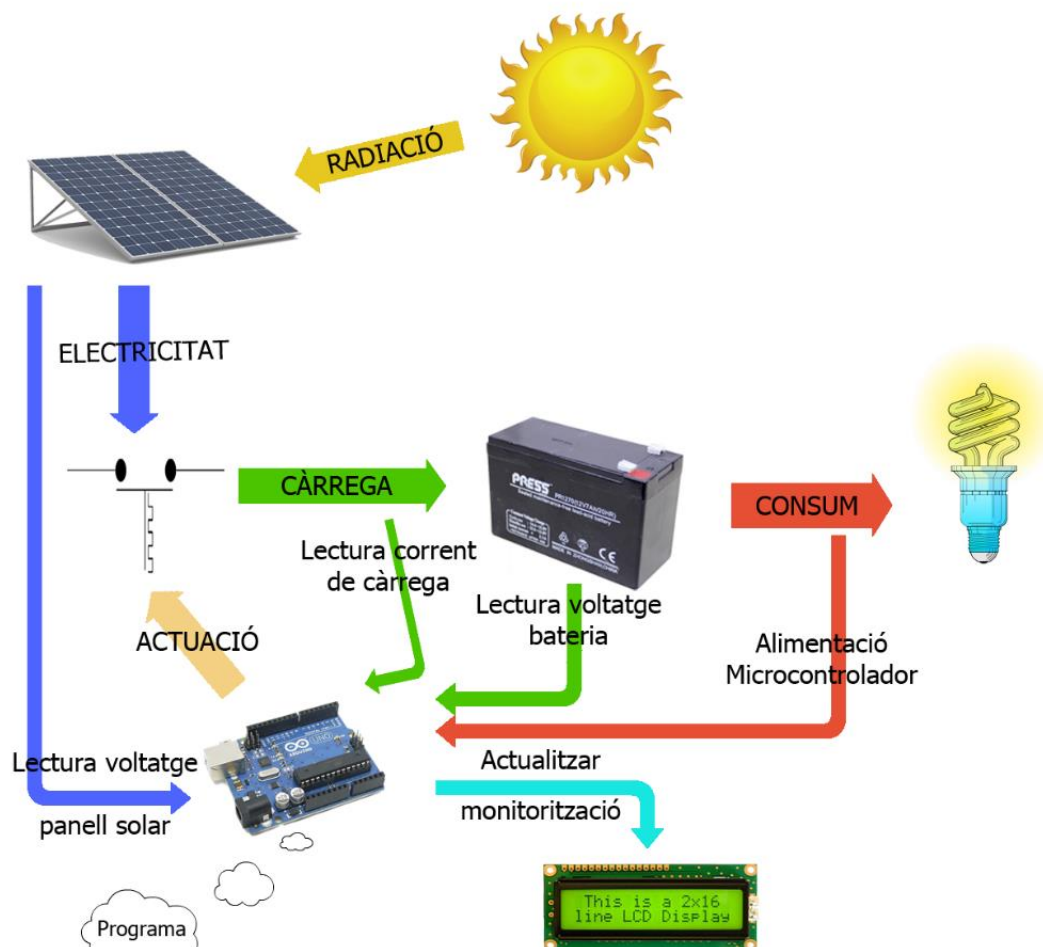


Figura 5. Imatge representativa del sistema

#### 4.2. Interfície de càrrega

La interfície de càrrega és una de les parts principal del projecte, ja que permet connectar la bateria i el panell solar entre ells i al mateix temps dóna l'opció al microcontrolador de llegir l'estat del sistema per poder prendre les decisions sobre quan i com actuar en el circuit.

##### 4.2.1. PWM

Aquest projecte utilitza una tècnica de PWM per gestionar la càrrega de bateries.

Un PWM (modulació per amplada de pols o "pulse-width modulation" en anglès) és una tècnica que modifica el cicle de treball d'un senyal periòdic (com podria ser un senyal quadrat o sinusoidal), ja sigui per transmetre informació a través d'un canal de comunicacions o, com és el cas d'aquest treball, per controlar la quantitat d'energia que s'envia a una càrrega<sup>[15]</sup>.

El cicle de treball (“*duty cycle*” en anglès) descriu la quantitat de temps que el senyal està en estat lògic alt (1) com un percentatge del temps total que aquest pren per completar un cicle.

Al canviar un senyal d’estat alt (1) a baix(0) d’una forma suficientment ràpida, la sortida es comportarà com un senyal analògic constant quan aquest està sent aplicat en un dispositiu.

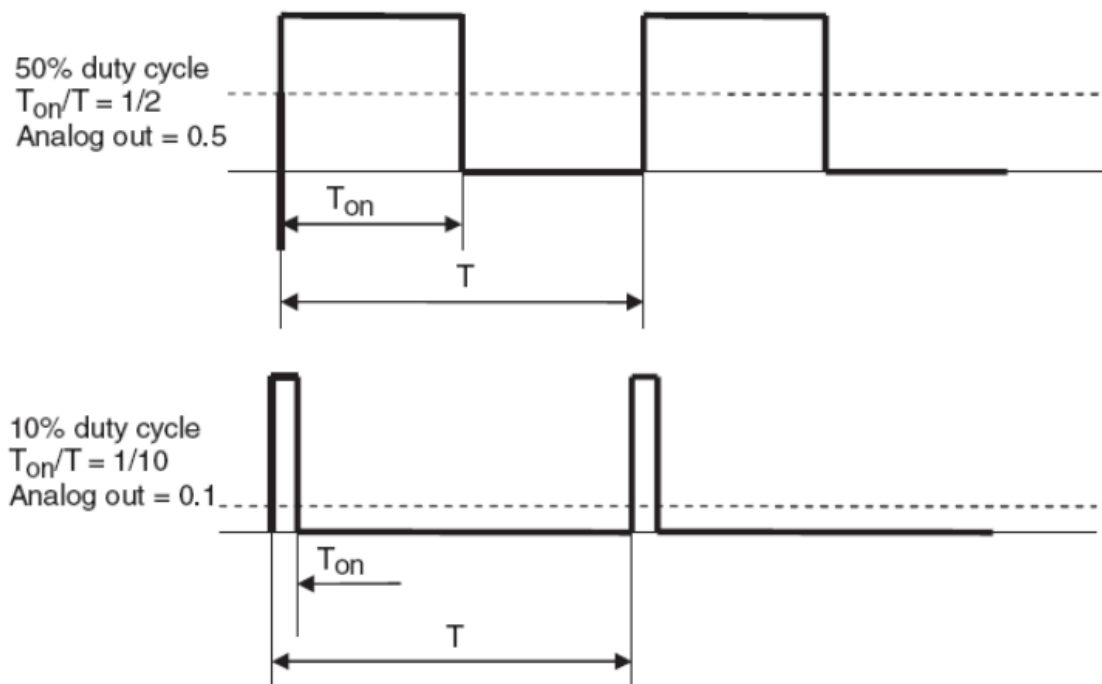


Figura 6. Exemples funcionament d'un PWM

En la Figura 6 es veuen dos imatges, la primera és un exemple d'un PWM on el seu cicle de treball és del 50%, com es pot veure, “ $T_{on}$ ” dura la meitat de temps del cicle complet “ $T$ ”. En la segona, el seu cicle és del 10%, i per tant, el seu temps de senyal alt (“ $T_{on}$ ”) equival a una desena part del període complet “ $T$ ”. Aquesta figura també mostra el valor que pren el senyal analògic quan s’aplica a un dispositiu (la línia discontinua). Aquest agafa la meitat del valor d’entrada de la font, en el primer cas, i la desena part en el segon.

Aquest projecte pretén utilitzar aquesta tècnica per carregar la bateria amb l’energia que es rep del panell solar.

Aquest procés és molt útil ja que augmenta la vida de la bateria, reduint els costos del sistema solar i els problemes d’eliminació de bateria. Augmenta la seva capacitat de reserva, ja que quan la temperatura és massa alta o el corrent no és suficient, aquest pot disconnectar la bateria i oferir la possibilitat de reduir la mida de la bateria per disminuir els costos del sistema i aprofitar millor l’energia solar dels panells [16].



Un altre motiu pel que es va triar aquest mètode de càrrega és que actualment la gran majoria de reguladors solars, quan el panell solar dóna un voltatge superior al que la bateria pot suportar, aturen l'arribada d'electricitat a les bateries fent així, que es perdin els moments més òptims del dia. Amb aquest procés, el PWM, simplement ha de reduir el valor energètic que fa arribar a la bateria al seu valor òptim, guanyant així, una gran quantitat de temps de càrrega respecte tots els altres reguladors.

#### 4.2.2. Estratègies de commutació

La connexió entre bateria i panell ha de poder ser interrompuda de forma controlada. Per aquesta funció es faran servir dos transistors per a controlar la connexió amb el microcontrolador, concretament el BUZ11 i l'IRF1405. Però hi ha dues possibilitats d'interrupció segons on s'instal·lin els transistors: tallar la connexió a terra (la massa) o bé tallar la connexió amb la font de voltatge (el positiu). Ambdues opcions ofereixen avantatges i inconvenients per tant s'han considerat totes dues en diversos prototips.

##### 4.2.2.1. Tallar la massa

Tot i que el més habitual per interrompre un circuit és fer-ho pel positiu, en el cas d'aquest projecte implica instal·lar el transistor entre V+ del panell i el V+ de la bateria. També és possible fer aquest tall pel costat de les masses, és a dir, muntar el transistor entre GND (o V-) de la Bateria i el GND del panell. Optar per aquesta segona opció permet tenir un maquinari més simple ja que no requereix treballar amb tensions altes.

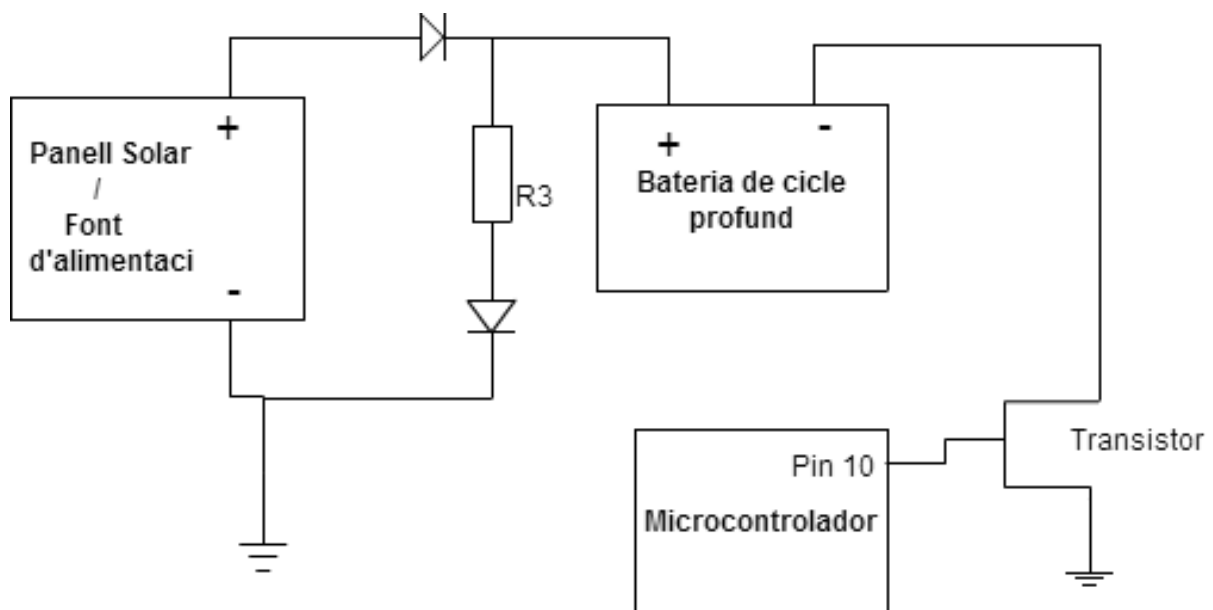


Figura 7. Exemples commutació per tall de la massa

Concretament en aquest model (Figura 7) es veu que l'energia elèctrica provinent dels panells arriba directament a la bateria a través dels pols positius. En cas que el microcontrolador volgués aturar el sentit d'aquest corrent, donaria l'ordre al seu Pin 10 i aquest tancaria la circulació del transistor, deixant la massa de la bateria "a l'aire" impedit que l'energia hi arribés.

La resistència R3 és necessària com a *pull up*, perquè garanteix que la capacitat paràsita de la gate del MOSFET es pugui descarregar amb el transistor tallat. Sense aquesta resistència no hi hauria sortida del corrent, que deixaria sempre saturat el transistor i, per tant, impediria el funcionament de la interfície de càrrega.

#### Resultats de tallar la massa

El gran problema d'aquest mètode de commutació és que, com que l'Arduino i la bateria tenen la mateixa massa, com es pot veure a la Figura 7, deixar la bateria sense massa implica també deixar-ne sense el microcontrolador, fent impossible el control de la càrrega. Perdre la massa apagaria el controlador i, en conseqüència, tot el sistema de càrrega del regulador. Per tant, es va abordar el problema per l'altre vesant: tallar el positiu.

#### 4.2.2.2. Tallar el positiu

Naturalment aquesta opció consisteix en situar el transistor a la branca del positiu per poder-la interrompre a voluntat. Cal tenir en compte que pels requeriments de voltatge que té aquest mètode també incrementen lleugerament els requeriments dels components.

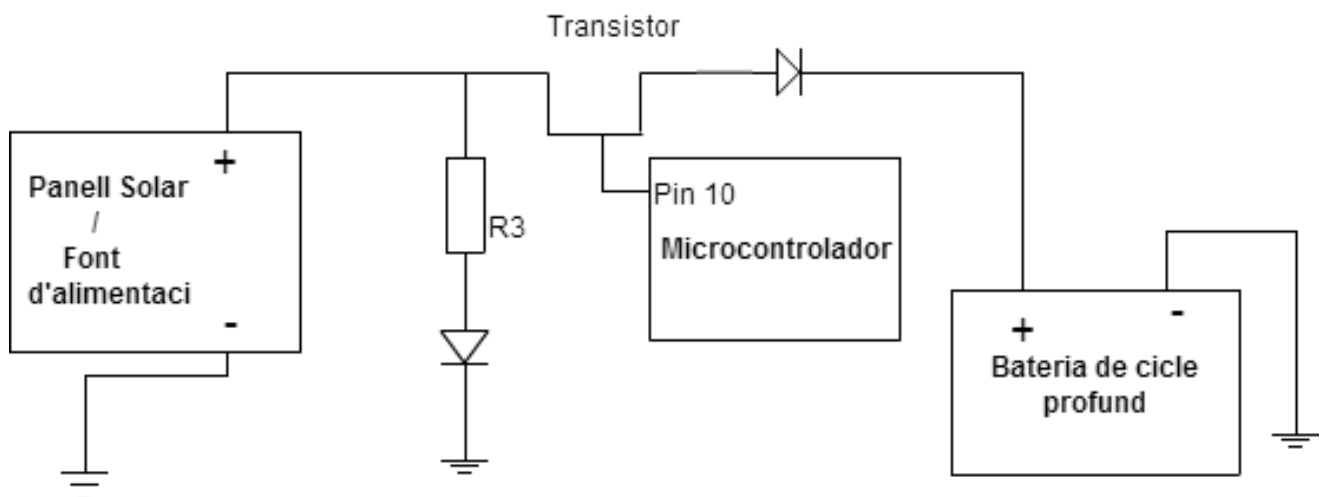


Figura 8. Exemples commutació per tall del positiu

Tot i que el funcionament bàsic és el mateix que en el model anterior, interromp la connexió entre el panell i la bateria de forma controlada per aconseguir la càrrega desitjada, en aquest cas es talla per la branca positiva i hi ha algunes diferències significatives en el connexionat per aconseguir els voltatges necessaris per controlar el transistor. En aquesta versió el transistor està per defecte saturat ja que rep el corrent directament del panell i l'Arduino pot descarregar-ne la *gate* donant una ordre que fa obrir el camí de descàrrega del transistor.

Tenint en compte la diferència "d'estat per defecte" va caldre modificar lleugerament el programari (negar el senyal que controlava el segon transistor) per fer funcionar el model.

### Resultats de tallar el positiu

Aquesta estratègia comporta una millora notable respecte el model anterior. Aquest mètode permet la interrupció de la connexió entre el panell solar i la bateria, però no desconnecta la font d'alimentació de l'Arduino, que com es veu a la Figura 8, és la bateria, mantenint les masses dels dos, tant del microcontrolador com de la bateria, sempre en comú.

#### 4.2.3. Divisor de tensió

La funció d'un divisor de tensió és "canviar l'escala" dels voltatges d'un circuit. En aquest cas, el microcontrolador només pot llegir voltatges entre 0V i 5V respecte el seu terra.

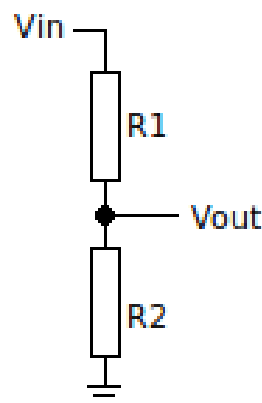


Figura 9. Divisor de tensió

#### 4.3. Tria dels components

Aquests són els components principals que s'han fet servir amb una breu explicació dels motius pels quals s'ha triat aquest model en concret. S'han obviat components simples com resistències i condensadors ja que independentment del fabricant, una resistència d'un valor òhmic concret es comporta sempre igual. La mateixa simplificació es pot aplicar als condensadors.

##### 4.3.1. Microcontrolador

El microcontrolador és "l'òrgan pensant" del projecte, el que pren les decisions de com s'ha d'actuar en cada moment i el que aplica l'algorisme i les accions de seguretat i monitorització.

S'ha optat per fer servir un **Arduino**<sup>[18]</sup>. La decisió de triar un producte d'aquest fabricant ha estat motivada principalment per dos motius: *el PWM i la seva gran comunitat d'usuaris*.

Un altre punt favorable ha estat que l'estudiant ja estava familiaritzat amb la seva arquitectura i eines de desenvolupament, donat que s'havia utilitzat en l'assignatura *Projecte d'Enginyeria de Computadors (PEC)*

Arduino incorpora, un seguit de funcions per calcular el PWM de forma automàtica, sense la necessitat de càlculs de temps en el qual el senyal ha d'estar alt (1) o baix (0). Aquest fet

simplifica força el treball del microcontrolador, que no haurà d'estar destinant una gran part dels seus recursos i temps a aquest càlcul i podrà dedicar-lo a altres funcions.

Pel que fa a la seva comunitat d'usuaris, tot i no disposar de dades concretes, segons el CEO de l'empresa Arduino SRL, encarregada de la producció d'aquests microcontroladors, mensualment s'hi uneixen al voltant de 100.000 persones. Només a la Xina ja hi ha milions d'usuaris que utilitzen aquest tipus de dispositiu de control<sup>[17]</sup>.

El model escollit ha estat l'**Arduino Uno**. S'ha triat aquest perquè, és el model estàndard i més complet de la gamma de controladors. També pel fet que consta de 6 ports que es poden fer servir com a sortides PWM, així com un seguit de pins digitals i analògics necessaris per a l'ús de la resta de dispositius utilitzats pel regulador de corrent.

El model Uno ja ve programat amb un "*bootloader*", el qual permet carregar-li un nou codi sense la necessitat d'un programador de maquinari extern. Simplement són necessaris els drivers i el programa del mateix fabricant. L'ordinador, per instal·lar el programa, es comunica amb el microcontrolador per connexió USB de tipus B – A. Un cop gravat el nou programa, l'Arduino pot ser alimentat mitjançant altres fonts, com una pila o una bateria, com és el cas d'aquest dispositiu. Aquest model pot treballar des de 5V quan està connectat via USB, i entre 6V i 20V, quan s'utilitza el port d'entrada de corrent "Vin". El més recomanat són valors entre 7 – 12V.

Mantenir un consum baix és imprescindible, ja que si la gestió de la càrrega consumeix massa energia, el resultat seria càrregues lentes o, fins i tot, descàrregues.

L'Arduino Uno disposa de 14 pins digitals, els quals poden actuar tant d'entrada com de sortida, i treballen a baixa potència. També incorpora 6 ports analògics, del A0 fins al A5. Aquests poden llegir voltatges des de 0 fins a 5 volts, i guardar-los en registres de 10bits.

La taula amb les especificacions tècniques d'aquest model es pot veure a continuació.

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Taula 3. Especificacions Arduino Uno

Aquestes característiques, tot i que no van ser fonamentals en la decisió, s'han aprofitat en el transcurs del projecte. L'ús que se n'ha fet està detallat en l'apartat del software.

#### 4.3.2. Sensor de corrent

Per mesurar el corrent que arriba a la bateria des del panell solar, és necessari un sensor de corrent ja que l'algorisme triat necessitarà saber aquest el valor.

Inicialment es va decidir fer servir un sensor ACS714, el qual proporcionava un ampli rang de valors cap els dos sentits (de 30A fins a -30A). Aquest mesurador de corrent és extremadament útil quan es tracta de consums o excessos de consum ja que la seva sortida produeix una tensió lineal proporcional al corrent que passa a través dels seus pins de mesura.

Tot i que l'ACS714 semblava molt adequat, finalment es va optar per un camí diferent, crear el sensor.

Per fer aquest sensor, s'ha connectat una resistència petita, de 1 Ohms, en el circuit que dóna càrrega a la bateria, i es mesura el voltatge, utilitzant l'Arduino, en cada punt d'aquesta resistència. Un cop tens la diferència de voltatge i el valor de la resistència, per calcular el corrent només cal aplicar la llei d'Ohm.

$$I = \frac{V}{R}$$

Figura 10. Llei d'Ohm

El fet que en aquest projecte s'utilitzin voltatges superiors a 5V, el màxim que l'Arduino pot suportar, ha fet necessari un divisor de tensió per poder utilitzar aquest sensor. Es pot veure en el diagrama d'aquest sensor a l'Annex A.

Aquest sensor, a més a més d'un divisor de tensió a cada banda de la resistència, porta incorporats uns condensadors, per tal que les lectures de voltatge en tot moment siguin constants i no es vegin afectades pel PWM, mostrant-nos 0 quan el transistor interromp la connexió.

#### 4.3.3. Sensor de temperatura

La temperatura de la bateria és un factor molt important en la vida d'aquesta. Les altes temperatures acceleren la corrosió i la degradació dels materials actius mentre que les baixes temperatures afecten la capacitat de proporcionar corrent de forma negativa, ja que tots els processos de corrosió interna es produeixen més lentament.

S'ha optat pel sensor DHT11, que és un dispositiu econòmic i de baix consum. Això és un punt fonamental ja que el regulador utilitzarà l'energia de la bateria i com més potència necessitin els dispositius, més difícil serà la seva càrrega. Si bé el DHT11 només llegeix temperatures entre 0° i 50° centígrads (°C) aquest regulador de càrrega solar no necessita anar més enllà d'aquest valor ja que la gran majoria de bateries a l'arribar als 40°C comencen a deteriorar-se. Un altre avantatge d'aquest sensor és que la seva sortida de dades és digital, la qual cosa suposa un gran avantatge respecte els sensors de tipus analògics, les fluctuacions de voltatge dels quals alteren la lectura de les dades.

#### 4.3.4. Transistors

Per poder aturar la circulació del corrent des del microcontrolador han estat necessaris dos transistors. Un connectat a l'Arduino, i un altre, de potència força alta, encarregat d'interrompre el corrent entre el panell solar i la bateria.

Els dos transistors escollits són respectivament el **BUZ11**, i el **IRF1405**. Aquests dos models són N-MOSFET i han estat triats perquè tenen un consum molt baix i perquè admeten corrents de pols molt alts (> 30A) la qual cosa és ideal per bateries d'alt amperatge. Un altre motiu de la seva elecció és que tant l'un com l'altre presenten una resistència interna pràcticament negligible (< 50mΩ) en saturació.

#### 4.4. Disseny inicial

La idea inicial d'aquest projecte era carregar la bateria amb les plaques solars i, si aquestes proporcionaven energia suficient, utilitzar l'energia restant per a dispositius externs, com una ràdio o una televisió,...

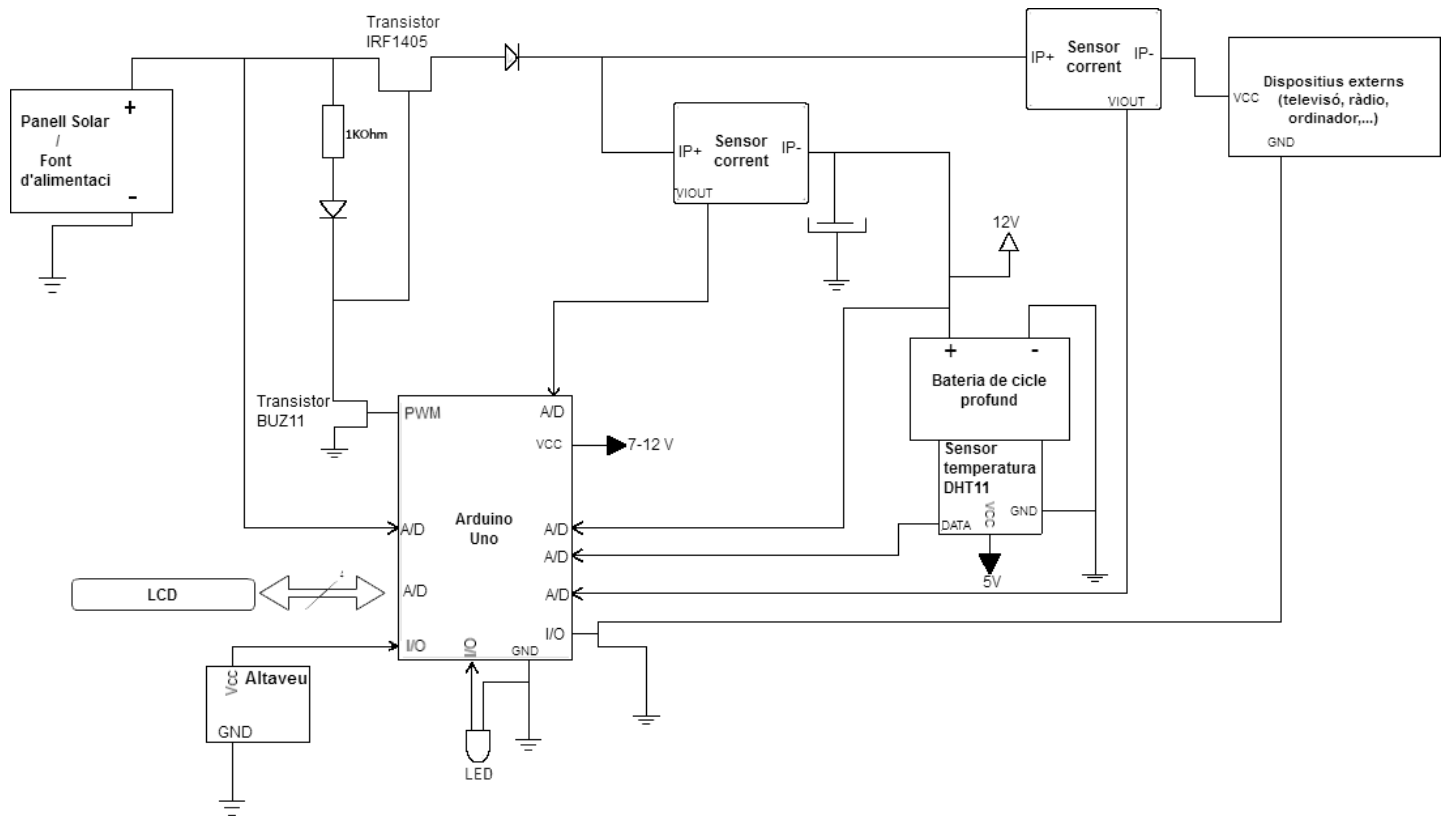


Figura 11. Diagrama de blocs inicial

En la Figura 11 es pot veure el diagrama de blocs que es va dissenyar per complir els objectius comentats anteriorment. Tot i que aquesta opció va semblar una bona idea inicialment, un cop dissenyat i revisat amb més atenció, van sorgir diversos problemes.

El principal problema d'aquest diagrama de blocs va ser l'intent de carregar dispositius externs amb l'energia sobrant del panell solar. La tècnica aplicada és la del PWM, com s'explica en el capítol previ, el transistor va deixant passar el corrent en determinats períodes de temps, i per tant els dispositius externs van rebent energia de manera intermitent fent que el seu funcionament correcte no sigui possible, ja que s'anirien apagant i encenent en funció d'aquests talls d'electricitat.

Tot això, sense tenir en compte que en el circuit s'haurien de fer servir un mínim de dues lectures de corrent, una per saber el que arriba a la bateria i l'altra per saber el que arriba a la resta de dispositius ja que al dividir-se el corrent seria impossible calcular el valor òptim necessari per a la bateria sense saber quin és el que arriba als dos llocs.

#### 4.5. Disseny final

Un cop detectats els problemes de la primera versió, s'ha modificat el plantejament, arribant al següent diagrama de blocs:

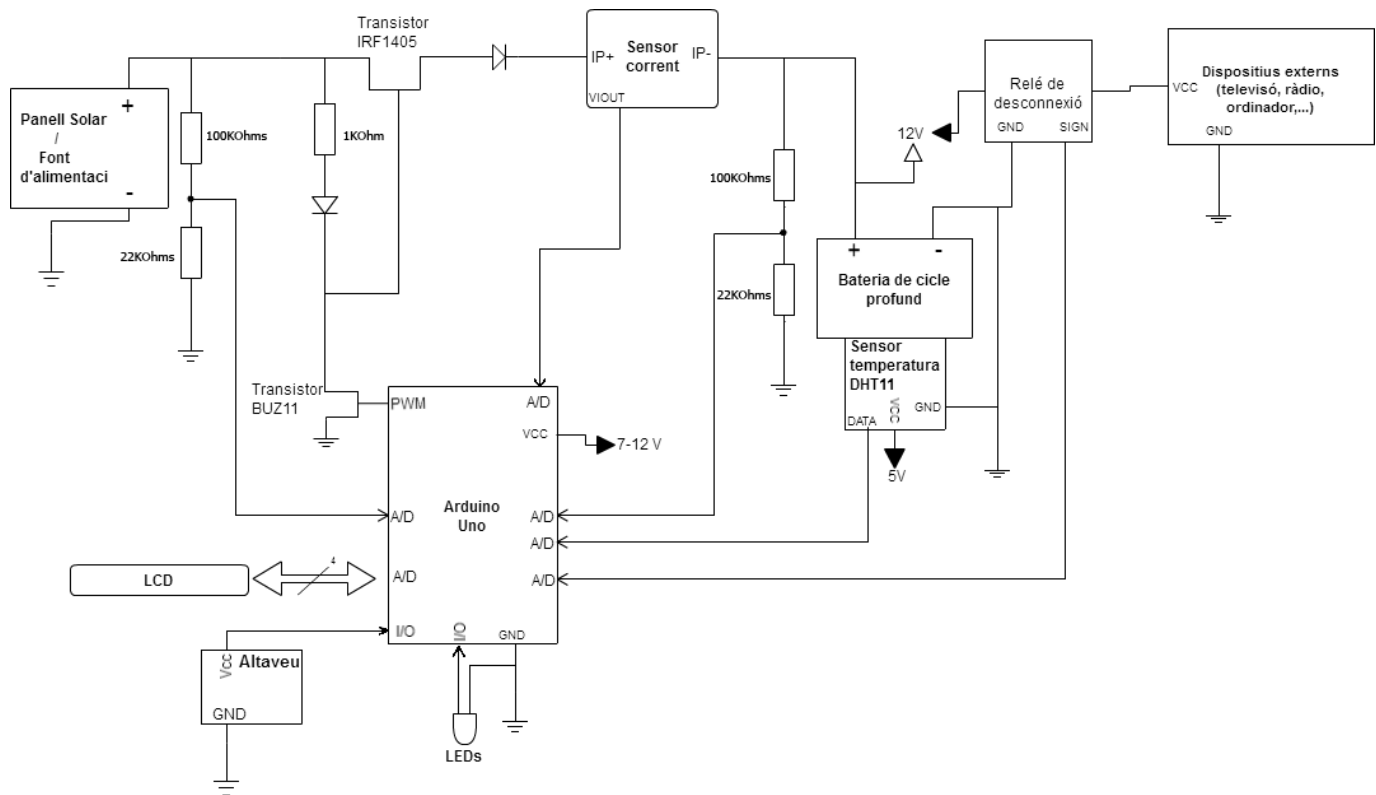


Figura 12. Diagrama de blocs final

L'Arduino està alimentat per la bateria, per tant el seus terres han de ser els mateixos i en conseqüència només pot mesurar tensions de fins a 5V respecte el terra de la bateria. Aquí és on entra en joc el divisor de tensió, ja que els panells solars treballen a un voltatge més alt que el que el microcontrolador pot suportar. Per al prototip d'aquest projecte es parteix de l'ús d'un panell solar de 12V el qual, en els moments més òptims del dia, pot produir fins a 25V, per tant, per a aquest divisor de tensió s'han utilitzat una resistència de  $R1 = 100K\Omega$  i una segona de  $R2 = 22K\Omega$ . Aquestes resistències han estat triades per que el voltatge que llegeix el microcontrolador pugui arribar fins a 27V aproximadament, i així deixar un marge d'error en cas que els panells solars tinguin un subministrament més alt de l'esperat.

En la Figura 12 es veu com els dispositius externs segueixen apareixent, la diferència entre aquesta figura i l'anterior és principalment el mètode de connexió i desconexió i d'on reben aquests últims l'electricitat.

Mentre en el primer diagrama de blocs es tenia en ment tallar la connexió entre el regulador i els dispositius externs per la massa, aquesta opció afegeix un relé el qual ho fa aturant la connexió pel positiu.

Com s'ha vist en l'apartat anterior, els dispositius externs no poden fer servir el corrent provinent del panell solar ja que s'estarien encenent i apagant constantment, per tant s'han



connectat directament a la bateria, de manera que quan es va carregant la bateria aquesta pot ser utilitzada. Amb aquest tipus de connexió entre la bateria i els dispositius externs, ja no és necessari tenir més d'un sensor de corrent, donat que només arriba corrent a la bateria des de la font d'alimentació solar.

Tant en la Figura 11 com en la Figura 12 es pot apreciar amb un triangle blanc l'energia que produeix la bateria, 12V, mentre que quan un dispositiu rep energia d'aquesta queda representat amb un triangle negre, com pot ser el cas de l'Arduino en les dues figures o del relé de desconexió en la segona.

## 5. Algorisme

A la Figura 13 es pot veure el temps de càrrega estàndard d'una bateria. En aquesta imatge es veu que una càrrega "no interactiva" de la bateria comença de forma força ràpida però a mesura que s'acosta al 20% final, la seva velocitat disminueix considerablement, fent que aquest 20% restant pugui arribar a tardar el mateix temps, o més, que el 80% previ.

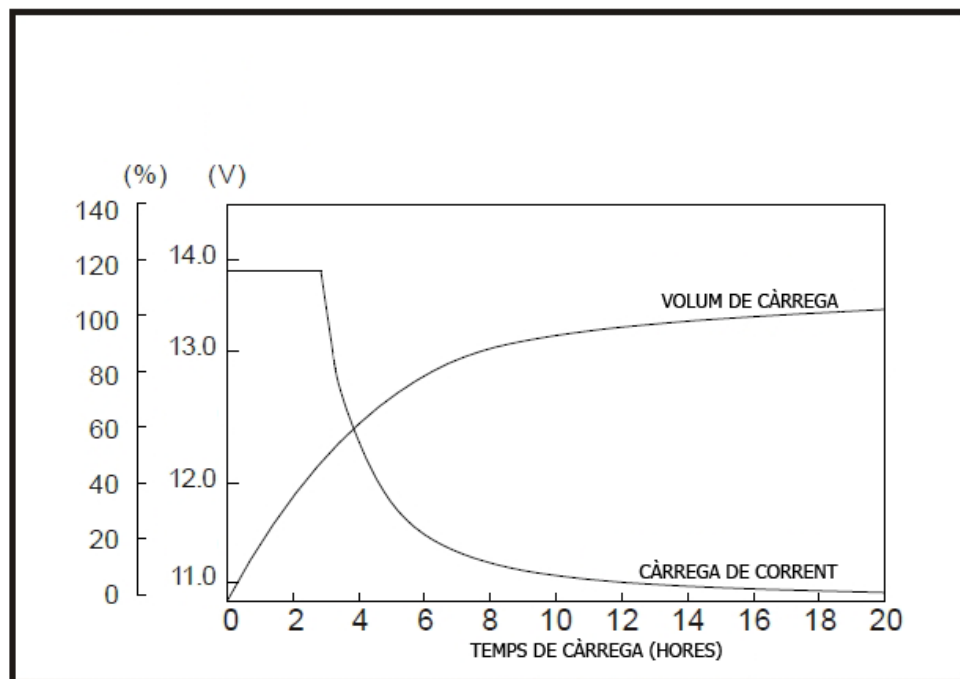


Figura 13. Algorisme de càrrega estàndard d'una bateria

### 5.1. Optimització de càrrega

El regulador de càrrega desenvolupat en aquest projecte controla el corrent proporcionat per les plaques solars, així com el seu voltatge, utilitzat per carregar les bateries, per tal que aquesta càrrega, a més a més, de tardar el menor temps possible, no afecti la vida útil de la bateria.

L'algorisme de càrrega és un conjunt de mesures i decisions de control (software), les quals són aplicades seqüencialment sobre els paràmetres elèctrics del sistema de càrrega (hardware) en un temps determinat, per aconseguir la càrrega d'una bateria.

Per tal de carregar la bateria en la seva màxima capacitat, de la forma més ràpida, i maximitzar la seva vida útil tant com sigui possible, l'algorisme utilitzat, s'anomena **algorisme de càrrega en cinc etapes**.

Assumint la descàrrega total d'una bateria, la seqüència de càrrega d'aquest algorisme segueix les cinc etapes següents:

### **1. Càrrega mínima**

Si el voltatge de la bateria és menor a un valor límit mínim indica que la descàrrega de la bateria és profunda o que existeixen una o més cel·les curtcircuitades, caldrà aplicar un mínim corrent per portar la bateria a un voltatge proper a la descàrrega normal.

Aquesta càrrega mínima en una bateria amb baix voltatge evita que els panells solars entreguin, de forma deliberada, corrents elevats a una cel·la curtcircuitada, impedit així les gasificacions en la cel·la danyada.

Ja que aquest regulador està pensat per l'ús de bateries de cicle profund principalment, s'espera que la descàrrega d'una bateria no es consideri profunda fins que resti aproximadament el 40% de la seva càrrega. Tot i així, quan l'usuari comenci a utilitzar el regulador podrà especificar quin és el percentatge exacte a partir del qual la descàrrega es considerarà profunda, perquè cada bateria podrà tenir un valor de descàrrega profunda diferent.

### **2. Càrrega màxima**

Una vegada que la càrrega mínima supera el límit mínim requerit de voltatge, especificat per l'usuari, l'algorisme passa a la fase de càrrega màxima. Durant aquesta etapa es proporcionarà el corrent constant màxim possible a les bateries (40% de l'amperatge/hora de la bateria), les quals recuperaran la major part de la seva capacitat, fins a un 80% de la càrrega, a mesura que el seu voltatge s'incrementi.

S'ha triat aquest valor perquè en aquesta etapa el percentatge de càrrega de la bateria és prou baix per que la bateria es carregui de forma ràpida, com es pot veure a la Figura 13, sense possibilitat de patir cap dany.

### **3. Sobrecàrrega controlada**

Un cop la fase de càrrega màxima ha finalitzat, la bateria hauria de tenir un 80% de la seva càrrega completa. Aleshores la bateria és sobrecarregada durant un període de temps on el carregador disminueix la seva intensitat, mentre aquesta continua absorbint energia.

En aquesta etapa el corrent de sobrecàrrega baixa força respecte la fase anterior. Això és degut a que, com la Figura 13 mostra, la bateria s'està acostant molt a la seva màxima càrrega i rebre molt corrent podria produir gasificacions degut al sobreescalfament de la bateria. Degut al efecte electroquímic produït per la bateria a mesura que aquesta va sent carregada les substàncies químiques van quedant saturades. S'ha establert per aquesta etapa que el millor corrent que pot rebre és el 20% del corrent total suportat.

Quan aquesta etapa, es dona per finalitzada quan aconsegueix arribar al 95% de la seva càrrega total.

#### 4. Equalització

L'equalització de bateries fotovoltaïques, té com a objectiu aprofitar el màxim de la capacitat de càrrega d'una bateria solar, allargant la seva vida útil, així com la seva eficiència i emmagatzement.

Aquesta fase s'utilitza per equilibrar el voltatge i la densitat de l'electròlit de cada cel·la, ja que quan una bateria està sent carregada, el sulfat que conté la placa de la bateria s'allibera cap a l'aigua formant àcid sulfúric que es queda en el plom de les plaques. Aquest sulfat s'anirà cristal·litzant i provocarà que la capacitat de la bateria solar es vagi reduint i fent-la cada cop menys eficient, fins que quedi inservible.

Durant el procés d'aquesta etapa, aquest sulfat que s'està cristal·litzant en les plaques de la bateria tendirà a combinar-se amb els electròlits, convertint-se així en àcid sulfúric un altre cop i precipitant-se cap a la part inferior de la placa.

A l'iniciar l'equalització la bateria ja té un 95% de la seva càrrega. Per accelerar la càrrega del 5% restant, el regulador proporcionarà una càrrega constant de valor reduït, al voltant del 10% de la seva intensitat.

#### 5. Flotació

En aquesta fase la bateria ja ha arribat al 100% de la seva càrrega.

La part final d'aquest algorisme s'utilitza per mantenir la capacitat de la bateria evitant la seva auto-descàrrega. En aquesta etapa s'aplica un corrent de flotació al voltant del 1% de l'amperatge/hora de la bateria per mantenir un voltatge constant, algunes dècimes per sobre del valor del voltatge de la bateria totalment carregada.

ALGORITME DE CÀRREGA DE 5 ETAPES

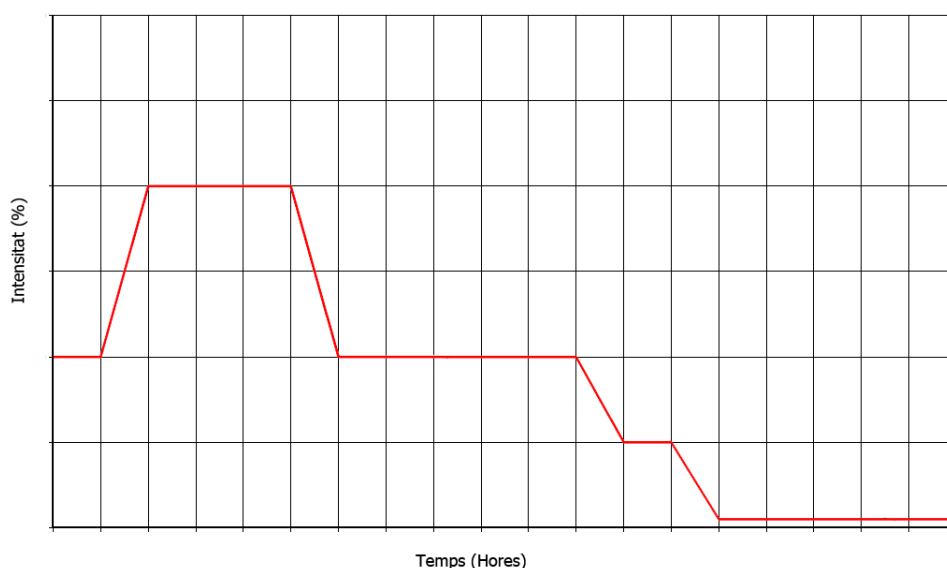


Figura 14. Algorisme de càrrega de 5 etapes

## 5.2. Protecció i monitorització

Aquest algorisme, a més a més d'aplicar unes mesures per optimitzar la càrrega de la bateria, també té un sistema de protecció així com un sistema de monitorització d'aquesta.

### 5.2.1. Protecció

Com s'esmenta en capítols anteriors, la protecció de la vida útil de la bateria és un objectiu principal d'aquest projecte.

Per preservar la seva vida útil el màxim possible, les funcions principals que han de tenir en compte són:

- Protecció per sobretensió en els panells solars.
- Protecció contra sobre-consum de corrent de càrrega.
- Desconnexió de consum per baixa tensió de bateria.
- Desconnexió per sobreescalfament de la bateria.

Amb la tècnica de càrrega esmentada al apartat *d'interfície de càrrega*, el PWM, les dos primeres funcions de protecció de la bateria ja queden satisfetes. Això és gràcies a la seva gestió a través del 'duty cycle', que d'aquesta manera en cas de rebre una sobretensió o un sobre-consum de corrent aquest s'adaptarà a aquesta pujada, augmentant o disminuint el seu pols, i fent que el corrent i/o tensió que arriba a la bateria segueixi sent l'òptim.

Per controlar que la bateria no arribi a una descàrrega molt gran i/o total, el microcontrolador revisa l'estat de descàrrega (o càrrega) en que es troba la bateria constantment. Quan la bateria arriba a un cert valor de descàrrega, el microcontrolador interrompt l'ús d'aquesta fins que no torna a tenir un percentatge de càrrega que es consideri segur per la vida útil de la bateria.

Com que cada bateria té un valor diferent de baixa tensió i de marge de seguretat per tornar a ser utilitzada, el regulador suggerirà un valor de 35% pel primer cas, i del 15% més pel segon, però en última instància serà l'usuari qui, amb les instruccions del fabricant, podrà escollir els valors més adequats d'aquests.

A més a més de poder escollir aquests valors, quan es connecta la bateria al regulador de càrrega solar, aquest demana un seguit de paràmetres més i l'usuari haurà de seleccionar els més adients per cada cas:

- Voltatge de la bateria
- Amperatge de la bateria
- Temperatura de sobreescalfament.

Aquestes valors són necessaris, no només per la protecció de la bateria sinó per la seva monitorització i el seu algorisme de càrrega. En el cas del voltatge de la bateria serviria per poder saber el percentatge de càrrega exacte, i en el cas del corrent (amperatge) per poder calcular el corrent òptim en cada fase.

El cas de la temperatura de sobreescalfament servirà perquè, amb el sensor de temperatura que porta incorporat, el regulador sàpiga quan la bateria està en el punt de sobreescalfament i pugui desconectar l'ús d'aquesta així com aturar la seva càrrega per evitar possibles danys.

#### 5.2.2. Monitorització

Pel cas de la monitorització els objectius eren els següents:

- Avís per baixa tensió de bateria.
- Avís de desconexió de bateria.
- Avís per sobreescalfament de la bateria
- Visualització de la tensió de bateria.
- Visualització de la tensió del panel solar.
- Visualització de la fase de càrrega (% càrrega).

Un seguit de llums led, una pantalla LCD de 2x16 i un altaveu han estat afegits al regulador de càrrega solar amb la finalitat de satisfer tots aquests objectius.

Per la monitorització s'ha fet ús de divisors de tensió, ja que per escalar el voltatge que es mostra per la pantalla LCD són necessaris per evitar que l'Arduino pugui malmetre's.

## 6. Desenvolupament

Un cop es va tenir un diagrama de blocs que satisfia totes les condicions inicials del projecte, com per exemple que permetés aplicar el PWM sense problemes fent arribar corrent a la bateria quan fos necessari i aturant-lo quan no, es van seguir els següents passos:

- Filtratge del PWM

Aquesta fase consistia en la investigació més profunda del PWM. El principal objectiu era veure com aquesta tècnica funcionava, les matemàtiques que comportava aplicar i el mètode per fer-lo funcionar correctament.

Un cop es va considerar assolit el coneixement necessari, es va passar a la part pràctica de la tècnica. Per mitjà de l'ús d'un generador de freqüència, la font d'alimentació i un transistor, es va procedir a veure com el corrent era filtrat.

- Carregar per PWM

Després d'aconseguir el filtratge del corrent de forma satisfactòria, i amb el circuit dissenyat i construït, es va comprovar que aquest mode de càrrega pogués ser aplicat a bateries d'un sistema d'energia solar.

El circuit final, la Figura 12, va ser connectat a la font d'alimentació per un punt i la bateria per l'altre. Utilitzant el generador de freqüència com a microcontrolador i, amb l'ajut del oscil·loscopi, va quedar demostrat que efectivament la bateria podia ser carregada mitjançant l'ús d'aquesta tècnica, ja que el voltatge llegit per la bateria tenia els pics corresponents al PWM aplicat. La mateixa bateria actua com a filtre d'aquests pics i no suposa una reducció de la seva vida, com podria ser si es connectés a un corrent o voltatge molt superior a la que permet en llargs períodes de temps.

- Afegir microcontrolador

Una vegada fetes les comprovacions de funcionament del PWM, es va fer la substitució del generador de freqüència pel microcontrolador, l'Arduino. L'objectiu d'aquest pas era trobar la manera de controlar l'arribada i l'aturada de corrent del panell solar a la bateria. El mètode va resultar ser força senzill, ja que Arduino té una funció que fa els càlculs per ell mateix.

```
void analogWrite(uint8_t pin, uint8_t value);
```

Mitjançant la crida d'aquesta funció el propi microcontrolador establia un PWM a través del pin [pin] d'aquest, i amb el valor [value], el qual ha d'estar comprés entre 0 (pel 0% de 'duty cycle') i 255 (pel 100% de 'duty cycle').

- Sensor de corrent i la seva variació

Un cop el microcontrolador ja podia regular el PWM de forma automàtica, es va passar a incorporar el sensor de corrent i una funció de presa de decisions, segons els resultats retornats per aquest.

Finalment es va decidir que el valor aplicat al PWM fos creixent o decreixent en funció de la diferència entre l'amperatge òptim i el real. També es va limitar aquest càlcul a un numero determinat d'intents, per evitar que el microcontrolador es quedi en aquest càlcul, bloquejant totes les altres comprovacions, com les de seguretat per exemple.

#### - Algorisme

Arribat al punt de tenir un mètode de càrrega decidit, comprovar que funcionés, assegurar que el microcontrolador fos capaç d'utilitzar la tècnica i decidir com es durien a terme els càlculs que determinarien la forma d'actuar d'aquesta, el següent punt ha estat definir l'algorisme.

Després d'investigar entre varies opcions es va optar per l'algorisme en cinc etapes, explicat anteriorment. Un cop decidit es va poder passar a la programació d'aquest.

### 6.1. Entorn de desenvolupament

#### 6.1.1. Desenvolupament hardware

Per la part del desenvolupament hardware la principal àrea de treball ha estat el laboratori del departament d'ESAI a l'edifici C5.

L'ús d'aquest laboratori ha estat fonamental pel desenvolupament del regulador de càrrega, ja que allà es disposava d'equipament electrònic imprescindible, que d'altra forma hagués suposat un cost addicional. El material principal que s'ha fet servir ha estat:

#### - Font d'alimentació:

Per tal d'evitar l'ús de les plaques solars durant el desenvolupament, ja que això limitaria molt l'espai i sobretot les hores de treball, s'ha optat per utilitzar les fonts d'alimentació d'aquest laboratori. Això, a més a més de donar llibertat pel que fa a les hores i l'espai de treball, també ha ajudat a comprovar tot tipus de situacions, on el voltatge o el corrent poden variar per tal de veure si el microcontrolador reacciona de forma adequada.



Figura 15. Font d'alimentació

#### - Oscil·loscopi:

L'oscil·loscopi ha servit per veure quin era el pols de PWM que estava sent aplicat per l'algorisme, així com a quina tensió estava en tot moment el circuit. Això ha estat extremadament útil per veure el funcionament d'aquesta tècnica, i per detectar errors en situacions puntuals.



Figura 16. Oscil·loscopi



- Multímetre digital:

Aquesta eina s'ha fet servir sobretot pel desenvolupament físic del circuit, ja que detectar el correcte voltatge i amperatge en cada part del circuit hagués estat molt més complicat de fer d'altra manera. A l'hora de fer el sensor de corrent manual també ha ajudat a comprovar els valors que l'Arduino estava calculant.



Figura 17. Multímetre digital

### 6.1.2. Desenvolupament software

Per altra banda, tot i que gran part del seu desenvolupament s'ha hagut de dur a terme en el laboratori del departament d'ESAI, una altra part també s'ha pogut fer per separat, per exemple la codificació de la pantalla LCD o la detecció de premsa de botons, ja que les eines emprades no aportaven cap informació nova.

Respecte el programari a utilitzar en aquest projecte s'ha optat pel següent:

- Arduino IDE

És l'entorn de desenvolupament habitual per treballar amb productes Arduino el proporciona el fabricant de forma gratuïta. Ofereix exemples específics i generals de diverses funcionalitats d'aquest microcontrolador, com funcions generades per desenvolupar codi, per gravar-lo al microcontrolador i per veure les comunicacions a través d'un port sèrie, entre aquest i el programa. També té un gran nombre de biblioteques preparades per ser referenciades i utilitzades.

Un altre motiu pel que es va triar aquesta opció és perquè ja s'hi estava familiaritzat, gràcies a altres projectes desenvolupats amb aquest mateix microcontrolador en algunes assignatures de la Universitat.

### 6.2. Parts del programari

Es poden diferenciar 2 parts del programari desenvolupat: les biblioteques dels perifèrics i el programa de control.

#### 6.2.1. Biblioteques de perifèrics

Com s'ha esmentat en el punt anterior, Arduino proporciona un gran nombre de biblioteques, i el fet de que tingui una comunitat d'usuaris tan gran facilita molt a l'hora de buscar solucions a problemes de codificació.

Les biblioteques que s'han utilitzat han estat aquestes:

### LiquidCrystal I2C

L'LCD és la principal via de comunicació amb l'usuari. La majoria de pantalles LCD necessiten fer arribar els 16 pins que tenen a l'Arduino per tal de comunicar-s'hi. La pantalla que s'ha fet servir per aquest regulador però, utilitza un mòdul I2C de connexió amb l'Arduino, fent així que només siguin necessaris 4 pins (5V, GND, SCA i SCL). Aquesta biblioteca és l'encarregada d'aquesta comunicació.

Tot i que aquesta llibreria proporciona un seguit de funcions molt útils, l'estudiant n'ha creat d'altres de crida a aquesta biblioteca per facilitar-ne encara més l'ús.

### Capçaleres

```
LiquidCrystal_I2C(uint8_t lcd_Addr, uint8_t En, uint8_t Rw, uint8_t Rs, uint8_t d4, uint8_t d5,  
uint8_t d6, uint8_t d7, uint8_t backlighPin, t_backlighPol pol);
```

Funció constructora de l'objecte LCD. Segueix la descripció donada pel fabricant pel procés. Inicialitza la classe i defineix l'adreça I2C [Addr] de la pantalla. Aquesta funció no inicialitza l'LCD, només crea l'objecte. També estableix el pin de connexió[En], de lectura/escriptura [Rw], el de reinici [Rs] i els pins per on rebrà les dades [d4, d5, d6, d7]. Per ultima instància determina el pin de la llum de la pantalla [backlighPin] i el seu estat [pol] d'encès o apagat.

```
void begin(uint8_t chars, uint8_t row);
```

Inicialitza l'LCD i determina el número de files [row] i de caràcters [chars] que hi caben en cada fila de la pantalla.

```
void clear();
```

Esborra tot el que es mostra a la pantalla i posiciona el cursor a la posició (0,0) de l'LCD (a dalt a l'esquerra).

```
void createChar(uint8_t location, uint8_t charmap[]);
```

Crea un caràcter personalitzat per mostrar a la pantalla. La majoria de chips només suporten 8 caràcters de 5x8 píxels, per tant, aquest mètode està limitat a 8 possibles caràcters. Utilitza un vector de 8 posicions [charmap] per especificar el caràcter. Això només crea el caràcter a la posició de memòria [location], però no el mostra.

```
void write(uint8_t value);
```

Escriu un caràcter amb valor [value] creat amb la funció anterior 'createChar' a la posició actual del cursors.

```
void setCursos(uint8_t col, uint8_t row);
```

Posiciona el cursor de l'LCD a la columna [col] i la fila [row] on el següent caràcter serà escrit.

```
void print(uint8_t data, uint8_t base);
```

Mostra el valor [data] per la pantalla. Aquest valor pot ser 'char', 'byte', 'int', 'long' o 'string'. La variable [base] és opcional, i serveix per indicar la base numèrica en la que s'ha d'imprimir el número, ex: 'BIN' per binari o 'DEC' per decimal.

### DHT

Com s'ha comentat en l'explicació d'aquest sensor, en la 'tria dels components', per preservar la vida útil de la bateria el màxim possible, aquest juga un paper força important.

Les baixes temperatures només alenteixen la capacitat de càrrega de la bateria, però no afecten la seva vida útil, per tant, només s'utilitzarà el sensor de temperatura per evitar temperatures elevades.

### Capçaleres

```
DHT(uint8_t pin, uint8_t type);
```

Funció constructora de l'objecte Sensor de Temperatura. Segueix la descripció donada pel fabricant pel procés. Inicialitza la classe i defineix el pin pel qual rebrà i enviarà la informació [pin] i el tipus de sensor que és [type], ja que hi ha diversos sensors d'aquesta família que utilitzen la mateixa llibreria. Es tracta del DHT11.

```
void begin();
```

Inicialitza l'objecte sensor de temperatura.

```
float readTemperature();
```

Llegeix la temperatura, en graus Celsius (°C) per defecte, i retorna el valor en forma de número decimal (o float).

### 6.2.2. Programari

El programa desenvolupat consta de dos parts, la inicialització i el control de càrrega de bateria o programa principal.

La inicialització (o setup en anglès) es crida una sola vegada a l'inici del programa. Aquesta funció serveix per establir els pins i els valors inicials, correctes, perquè el microcontrolador sàpiga per a que serveixen cadascun, així com els objectes dels sensors o dispositius externs utilitzats, entre d'altres.

```
void setup() {  
    initialization_pins();           //Inicialització dels pins utilitzats a l'Arduino  
    lcd.begin(16,2);                //Inicialització de la pantalla LCD (2x16)  
    dht.begin();                    //Inicialització del sensor de temperatura  
    updatePWM(MINIMUM_DUTYCYCLE);  //Posar el PWM a 0 % inicialment  
    updateUsageOfBattery(false);    //No es permet l'ús de la bateria inicialment  
    powerToBattery = false;         //Indicar càrrega de bateria no començada  
    initialization_bat();            //Inicialització de les propietats de la bateria  
    percentage = 0;                 //Inicialització de variable global pel codi principal  
    batteryV = 0;                   //Inicialització de variable global pel codi principal  
    panelV = 0;                     //Inicialització de variable global pel codi principal  
}
```

El codi principal consisteix en un bucle infinit de crides a les funcions de control de l'algorisme de càrrega interactiu, així com a les actualitzacions de pantalla per informar l'usuari del estat actual del procés.

```
void loop() {  
    batteryV = readVoltageBattery(); //Llegim el voltatge de la bateria  
    panelV = readVoltagePanel();     //Llegim el voltatge del panell solar  
    percentage = getbatteryPercentage(batteryV); //Calcular el percentatge de la bateria  
    applyAlgorithm(percentage);      //Aplicar l'algorisme interactiu creat  
    lcd.clear();                     //Netejar la pantalla LCD  
    lcdStatus(percentage,panelV, batteryV); //Actualitzar l'estat que es mostra per LCD  
    ledStatus(percentage);           //Actualitzar l'estat que es mostra pels leds  
    delay(5000);                     //Esperar 5 segons per tornar a començar  
}
```

El codi complet del programa es troba als annexos. A continuació s'ofereix el diagrama de flux de l'aplicació.

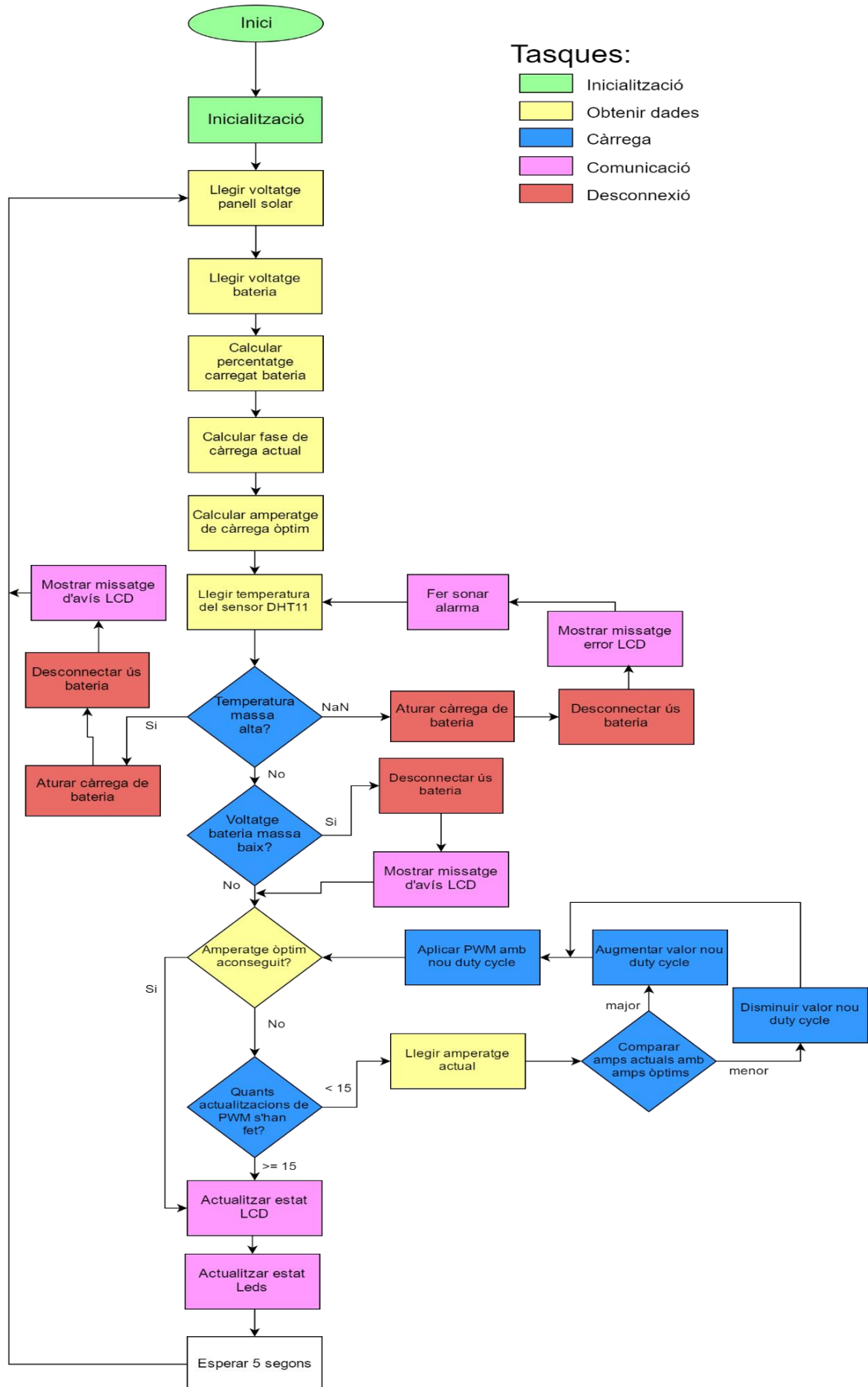


Figura 18. Diagrama de flux general

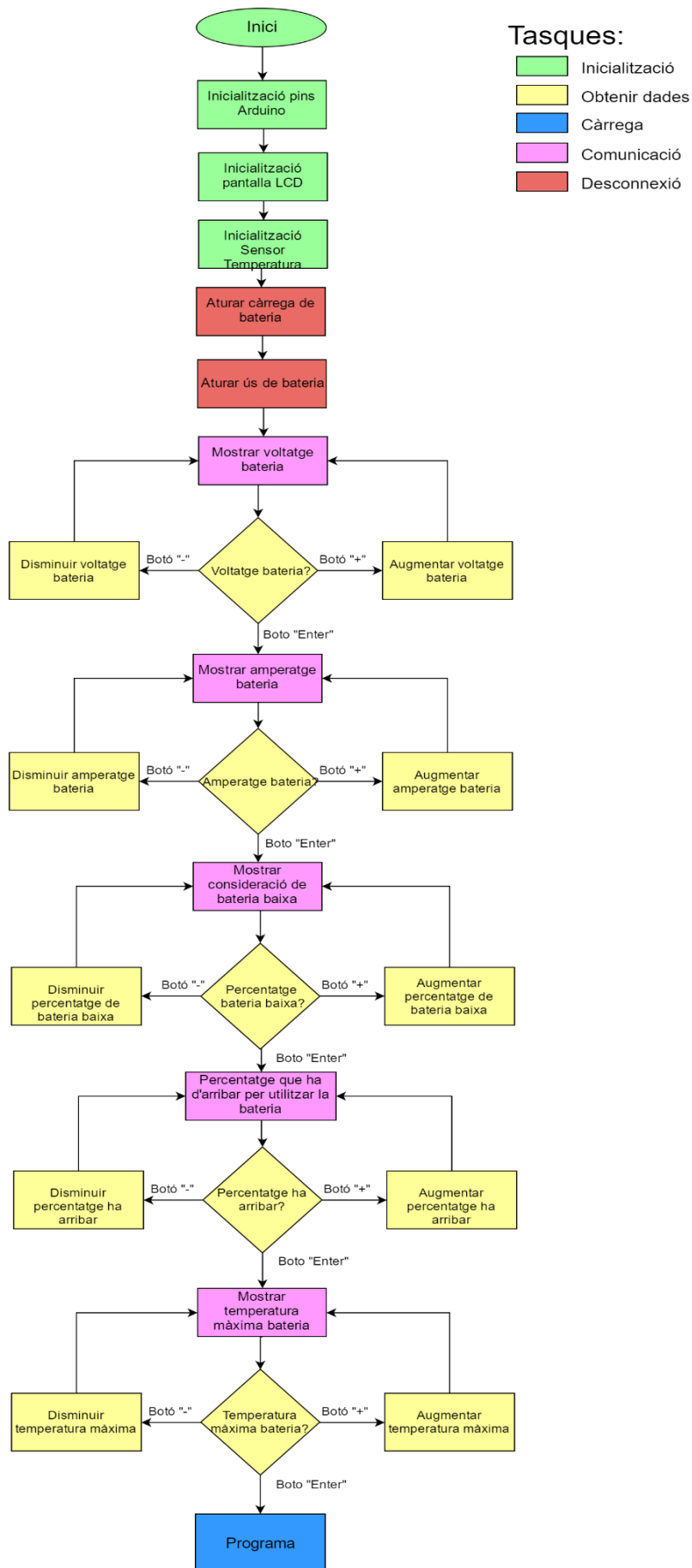


Figura 19. Diagrama de flux inicialització

En el primer diagrama de flux, Figura 18, es veu el funcionament global de tot el projecte. Aquest engloba totes les funcions d'optimització de càrrega, de protecció i de monitorització de la bateria, definides en els objectius, així com quin era l'ús de tots els sensors considerats necessaris pel treball.

Durant l'execució del programa es fan diverses lectures de voltatge. Aquestes lectures serveixen per:

- Poder mostrar quin és el voltatge que s'està rebent en el panell solar.
- Poder mostrar quin és el voltatge actual de la bateria, i per conseqüència, quin és el percentatge de bateria carregat.
- Per saber el corrent que està arribant a la bateria.

La lectura de voltatge pel sensor de corrent és fa un número major de vegades que la feta per determinar el percentatge carregat, ja que a mesura que es va variant el pols del PWM, el corrent varia i s'ha d'anar controlant si s'ha arribat a l'amperatge desitjat.

Per tal de calcular quin és el percentatge de bateria carregat, s'utilitza un vector d'estats de càrrega de la bateria, en el qual es determina per cada voltatge quin és el seu percentatge de càrrega corresponent.

Un cop s'obté el percentatge, es comença a aplicar l'algorisme interactiu. En primer lloc s'estableix en quina de les fases de l'algorisme s'està, i a quin amperatge de càrrega s'ha d'arribar.

Un cop se sap aquest valor, i abans de prosseguir a obtenir-lo, es fan les comprovacions de seguretat següents:

- Es comprova que la temperatura de la bateria sigui correcte. En el cas de que no fos així, s'aturaria l'algorisme de càrrega i l'ús de la bateria per tal de reduir aquest sobreescalfament. Aquesta funció també verifica que el sensor de temperatura funcioni correctament i, en el cas de no ser així, s'activa una alarma sonora per fer saber a l'usuari del regulador que hi ha un problema amb el sensor.
- Es comprova que encara es pot seguir fent ús de la bateria per dispositius externs, comparant el percentatge carregat de la bateria amb l'establert per l'usuari a partir del qual una descàrrega es considera profunda. Tant si és pot fer ús de la bateria com si no, aquesta es seguirà carregant.

Com s'explica quan es parla de les bateries de cicle profund, aquestes tenen una gran capacitat de descàrrega, però una de massa profunda podria afectar a la seva vida, per tant, el regulador ha de comprovar si es pot seguir fent ús de la bateria.

Un cop fetes totes les comprovacions, el regulador es dedica a canviar el pols del PWM, augmentant-lo o disminuint-lo, fins a arribar al corrent desitjat, o fins que s'han fet 15 intents per arribar-hi. Aquesta última comprovació serveix com a mesura de seguretat, perquè l'algorisme no és quedi bloquejat intentant trobar un corrent òptim durant un període massa llarg de temps oblidant així les comprovacions de temperatura i voltatge.

Quan s'arriba al corrent òptim o al número d'intents màxim, s'actualitzen tant la pantalla LCD com els leds que mostren l'estat de càrrega actual.

Abans de tornar a començar tot aquest procediment, el microcontrolador s'atura durant 5 segons per tal que les configuracions prèvies puguin ser aprofitades. S'entén que 5 segons, en les capacitats de treball del regulador són suficients per assolir un bon resultat.

També s'ha considerat necessari fer el diagrama de flux de la inicialització del programa, Figura 19, ja que aquesta no només consistia en la creació d'objectes o establiments de pins a utilitzar. Aquest procés també estableix:

- Quin tipus de bateria està sent connectada (6V, 8V, 12V, etc....). Aquesta comprovació es necessita per saber quins seran els estats de càrrega de la bateria.
- Amperatge de treball de la bateria. Hi ha diferents tipus de corrent per cada bateria i, per tal d'aplicar l'algorisme és necessari saber aquest valor.
- Percentatge en el qual ja es considera que la bateria està en estat de descàrrega profunda.
- Percentatge en el que es considera segur tornar a utilitzar la bateria amb dispositius externs. Aquest valor és necessari perquè si, per exemple, desconnectem l'ús de la bateria al 35% en el moment que es carregui un 1% més, el regulador tornarà a habilitar l'ús de la bateria i pot provocar que s'entri en un bucle difícil de sortir, ja que no hi hauria temps suficient perquè la bateria recuperi part de la seva càrrega.
- La màxima temperatura que pot suportar la bateria abans que la seva vida útil es vegi afectada.

També cal destacar que fins que la bateria no està configurada, no es permet que corrent arribi a la bateria ni el seu ús amb altres dispositius. Aquestes mesures es prenen per evitar fer malbé la bateria amb corrents alts abans que es pugui configurar, ni que aquesta pugui patir una descàrrega profunda.

S'han omès alguns detalls per facilitar la comprensió dels diagrames.



## 7. Gestió econòmica

### 7.1. Identificació de costos

És un recull de tots els materials utilitzats en la globalitat del projecte. Aquest conjunt ha estat utilitzat per poder calcular el cost general del projecte, així com per facilitar l'anàlisi econòmic fet posteriorment.

#### 7.1.1. Recursos materials

- A. HP Envy 15 Notebook PC
- B. Font d'alimentació
- C. Dispositius electrònics (transistors, resistències, leds, sensors, etc...)
- D. Microsoft Windows 10 Professional
- E. Material del laboratori d'ESAI
- F. Microsoft Office 2013
- G. Smartsheet
- H. Dropbox
- I. Google Drive
- J. Material de connexió (cables, pinces, etc...)
- K. Bateria de cicle profund
- L. Microcontrolador

#### 7.1.2. Costos directes per activitat

Activitat	Recursos
<b>Planificació del projecte i viabilitat</b>	
Abast i contextualització	A, D, F, I
Planificació	A, D, F, G, I
Gestió econòmica i sostenibilitat	A, D, F, I
Presentació preliminar	A, D, F, I
Documentació final i presentació	A, D, F, G, I
<b>Anàlisi i disseny</b>	
Definició dels requisits i objectius	A, B, C, D, H, I, J, K, L
Disseny hardware	A, B, C, D, E, H, J, K, L
Disseny software	A, C, D, E, H
Disseny sistema de verificació	A, B, C, D, E, H, I, K, L
<b>Fase de Desenvolupament</b>	
Elecció de dispositius externs	A, C, D, E, L
Interfícies d'adaptació	A, B, C, D, I, K, L
Memòria de programa	A, C, D, I, K, L
Diagrama de blocs	A, B, C, D, I, J, K, L
Elecció de microcontrolador	A, B, C, D, I, J, K, L
Estudi entorn de treball	A, B, C, E, I, J, K, L
Implementació hardware i software	A, B, C, D, E, H, I, J, K, L
Verificació sistema	A, B, C, D, E, H, I, J, K, L
<b>Etapla final</b>	
Documentació	A, D, F, G, I
Presentació	A, D, F, I

Taula 4. Utilització de recursos per activitat del Gantt

### 7.1.3. Costos indirectes

Les despeses generals i impostos no es comptabilitzen en el cost del projecte ja que tot es desenvolupa dins de la infraestructura de la Universitat. En cas de voler calcular-ho implicaria que s'hauria d'extreure una part de la despesa de l'edifici, on es desenvolupa el treball, que sigui atribuïble al projecte.

### 7.1.4. Amortitzacions

#### 7.1.4.1. Recursos materials

A continuació es detallen els costos associats als recursos materials. Cada recurs inclou la corresponent amortització en funció del temps de vida previst. L'amortització es calcula tenint en compte que el coeficient màxim permès pel Govern d'Espanya, és del 25%.

La fórmula del càlcul aplicat és:

$$\text{Amortització} = \frac{\text{Preu}_{\text{producte}} \times 0,25 \times \text{Hores}_{\text{projecte}}}{\text{Anys}_{\text{vida}} \times 365}$$

Recurs	Preu	Temps de vida	Amortització
<b>HP Envy 15 Notebook PC</b>	950€	4 anys	73.20€
<b>Font d'alimentació</b>	0€	2 anys	0€
<b>Dispositius electrònics</b>	118.71€*	2 anys	26.43€
<b>Microsoft Windows 10 Professional</b>	0€**	4 anys	--
<b>Material del laboratori d'ESAI</b>	0€	--	--
<b>Microsoft Office 2013</b>	80€	3 anys	8.22€
<b>Smartsheet</b>	0€***	1 any	--
<b>Dropbox</b>	0€	--	--
<b>Google Drive</b>	0€	--	--
<b>Materials de connexió</b>	8.50€	1 any	3.78€
<b>Bateries cicle profund</b>	25€	4 anys	5.56€
<b>Microcontrolador</b>	2.39€	2 anys	0.53€
<b>Total</b>	1184.6€		117.72€

Taula 5. Costos i amortitzacions dels recursos materials

\*S'ha fet una suma del preu total de tots els dispositius utilitzats. Pels preus concrets dels dispositius electrònics més importants utilitzats, seguir la secció de cost unitari.

\*\*El sistema operatiu *Windows 10* està disponible gratuïtament a la pàgina de Microsoft.

\*\*\**Smartsheet* ofereix un any gratuït d'ús, el qual ha estat temps suficient per fer el diagrama de Gantt i tots els seus càlculs.

#### 7.1.4.2. Recursos humans

A la següent taula es detalla la dedicació final en hores i el cost dels diferents rols que participen en el projecte.

Rol	Hores dedicades	Preu per hora	Cost total
Cap de projecte	30 hores	50€/hora	1500€
Enginyer de suport a la investigació	70 hores	30€/hora	2100€
Becari d'investigació	550 hores	10€/hora	5500€
<b>Total</b>	<b>650 hores</b>		<b>9100€</b>

Taula 6. Costos dels recursos humans

## 7.2. Estimació de costos

A continuació es mostra el cost total del projecte. El cost dels recursos materials està en funció de les hores treballades (inicialment es van preveure 450 hores pel projecte).

Tant en els imprevistos com en les contingències es comptabilitza la substitució/adquisició de nous dispositius, per alguna errada o avaria dels escollits, de fins un cost total de 90€. Tanmateix s'assumeix un cost extra de 100€ per a la possible substitució d'una peça a l'ordinador utilitzat pel desenvolupament.

Amb aquestes dades es va fer una estimació inicial del projecte. Aquesta es veu reflectida en la següent taula:

Concepte	Cost
Recursos materials	1184.6€
Recursos humans	8100€
Contingències	190€
Imprevistos	100€
Total	9574.6€

Taula 7. Estimació inicial del cost total del projecte

Un cop finalitzat el projecte, i revisat els costos, el costos reals han estat aquests:

Concepte	Cost
Recursos materials	1184.6€
Recursos humans	9100€
Contingències	190€
Imprevistos	100€
Total	10574.6€

Taula 8. Cost total del projecte

La diferència entre l'estimació inicial i el cost real és deguda, al augment d'hores que ha suposat el projecte, respecte la planificació inicial, com ja s'estipula en apartats anteriors.

## 7.3. Control de gestió

Cada setmana, o cada dos setmanes, s'ha realitzat una reunió de seguiment i supervisió amb el tutor en la que s'analitzava l'estat del projecte per preveure necessitats futures i anticipar-s'hi, ja sigui per la necessitat d'adquirir nou material, detectar errors de disseny o fer modificacions en el plantejament del problema per millorar la solució del moment.

### 7.3.1. Desviaments

Es tracta de fer l'anàlisi de costos per estudiar les variacions i l'estat projecte en cada moment i així evitar les pèrdues pressupostàries.

Per tal de controlar els desviaments de les despeses s'ha aplicat un control de costos a mesura que el projecte anava avançant i aquests s'anaven produint. Aquest control ha consistit en comparar les despeses reals respecte les que han estat previstes o pressupostades.

Un exemple de les fórmules que s'han utilitzat és:

Desviaments totals:
oDesviaments per volum en Ventes = (Ventes reals – ventes pressupostades) * Marge estimat
oDesviaments per marge en Ventes = (Marge real – Marge estimat) * quantitat real de ventes
oDesviament total en mà d'obra = cost total estimat mà d'obra – cost total real mà d'obra
oDesviament total en matèria prima = cost total estimat matèria prima – cost total real matèria prima
oDesviament total ventes = total ventes pressupostades – total ventes reals
oDesviament total Costos Fijos = cost total fix pressupostat – cost total fijo real

Figura 20. Fórmules pel càlcul dels desviaments

Afortunadament, tot i les desviacions produïdes, sobretot degut a la durada del desenvolupament del projecte, en cap moment s'ha considerat que aquestes fossin prou grans com per que es considerés l'aturada del projecte.

#### 7.4. Cost unitari

Cal tenir en compte que hi ha dos conceptes molt diferents a valorar, l'un és el cost del desenvolupament del projecte, Taula 8, i l'altre el cost unitari de cada regulador, assumint que se'n cedeix gratuïtament el coneixement tècnic.

Concepte	Cost
Arduino Uno	2.39€
IRF1405	1.37€
BUZ11	1.55€
Pantalla LCD 2x16	10€
Relé de desconexió	1.27€
Altaveu	1.30€
Sensor DHT11	1.26€
Breadboard	2.08€
Altres components	7.50€
Muntatge	10€
<b>Total</b>	<b>38.72€</b>

Taula 9. Cost unitari del regulador de càrrega solar

S'han omès els costos del panell solar i de la bateria ja que a més a més de ser molt variables, no impliquen cap diferència sobre el del regulador.

Com es veu a la Taula 9 el cost unitari del regulador de càrrega dissenyat és de **38.72€**.

Aquest preu és molt assumible i més tenint en compte que es podria reduir si es produís de manera industrialitzada en una cadena de producció. Aquest preu és un indicador que permet considerar l'objectiu de desenvolupar una solució de baix cost ben aconseguit, sobretot si es té en compte que els reguladors actuals que utilitzen la tecnologia de PWM, tot i no portar un algorisme de càrrega optimitzat, tenen un preu aproximat de 70€.

## 8. Sostenibilitat

### 8.1. Dimensió econòmica

Des del punt de vista econòmic, un dels objectius del projecte consistia en maximitzar l'obtenció d'energia elèctrica a través d'una font alternativa, reduint així el cost en les factures d'electricitat mensuals. Un altre objectiu de pes era reduir el preu de futures instal·lacions elèctriques gràcies al baix cost del regulador de càrrega desenvolupat. Per tant, la inversió inicial que suposa tindrà com a resultat un estalvi econòmic general.

D'altra banda, algú amb més experiència podria haver dut a terme el projecte reduint-ne el temps de desenvolupament però no els recursos. No obstant, això implicaria un augment en els costos pel que fa a l'àrea dels recursos humans ja que la seva millor posició en el camp de la informàtica implica una major remuneració, fent que el cost final hagués acabat sent aproximadament el mateix.

Cal destacar que com major sigui la inversió a nivell de sensors, el sistema serà més precís i generalitzat tenint, però, una major despesa en maquinari. De totes maneres a llarg termini es preveu que la inversió inicial acabi generant beneficis.

### 8.2. Dimensió social

A partir de l'any 2015 sembla que la situació econòmica comença a remuntar després d'uns quants anys de recessió. A nivell català, espanyol i europeu hi ha certa inestabilitat política. És època de reestructuració de les formes de govern cap a nous models polítics i aquesta inestabilitat es veu reflectida en l'àmbit social sobretot en el distanciament entre classes altes i baixes.

Com s'ha dit en el capítol de la Introducció, l'objectiu del TFG és reduir el consum energètic mantenint la comoditat dels usuaris en l'habitatge. Aquesta finalitat afavoreix qualsevol persona, del nivell social que sigui, per l'estalvi que suposa, tant pel fet de no necessitar fer arribar instal·lacions elèctriques a determinats llocs (com seria una casa de muntanya, una masia o un refugi) com per la reducció en la despesa elèctrica.

A més a més, el seu àmbit d'acció no només es limita a l'habitatge, totes les administracions públiques o privades, centres de recerca, etc... poden aprofitar aquesta manera d'estalviar ja que l'autoconsum permet aquesta reducció en els costos.

Un altre efecte que produeix l'ús d'aquesta energia renovable és que no només es limita al ciutadà. El sistema de producció d'electricitat d'un país es podria veure revolucionat per aquesta tecnologia permetent reduir els costos de producció, afavorint així l'economia de l'estat. Això milloraria l'estat del benestar del país, sense haver de destinar tants recursos a produir electricitat i poder-los utilitzar per altres assumptes.

### 8.3. Dimensió ambiental

La Taula 4 detalla els recursos que s'utilitzen a cada etapa del projecte. El consum és només elèctric i tots els components són de baix consum, seguint la filosofia del projecte.

A llarg termini s'espera que hi hagi un estalvi major a la despesa generada. A més a més, tot el material, a excepció dels sensors específics, és aprofitable per a qualsevol altre generador.

Si bé els circuits elèctrics estan formats per materials que provenen de mines en les que es desconeix la situació real dels treballadors, el projecte compensa indirectament aquest factor ja que redueix el desgast d'altres components elèctrics allargant-ne la seva vida útil i per tant reduint els residus i millorant la sostenibilitat del planeta.

Aquest treball intenta contribuir a la reducció de la petjada ecològica que la societat actual estan deixant. Per mitjà de les energies renovables es pot reduir aquesta empremta, ja que l'aprofitament d'aquestes no va associat a un empitjorament del medi ambient, més aviat s'associa amb una adaptació positiva d'aquest. Es considera positiva perquè per extreure l'energia d'aquests recursos no s'ha d'alterar el medi, simplement aprofitar els seus avantatges. Aquesta és la finalitat principal d'aquest regulador, poder fer el màxim ús possible del que disposem sense tenir la necessitat d'alterar el medi ambient.

### 8.4. Puntuació

Sostenible?	Econòmica	Social	Ambiental
Planificació	Viabilitat econòmica	Millora qualitat de vida	Anàlisi de recursos
Valoració	10	10	10
Resultats	Cost final versus previsió	Impacte en l'entorn social	Consum de recursos
Valoració	9	10	10
Riscos	Adaptació a canvis d'escenaris	Danys socials	Danys ambientals
Valoració	0	0	0
Valoració total	59		

Taula 10. Matriu de sostenibilitat del TFG

En el camp de la sostenibilitat aquest projecte es considera 100% sostenible, degut a que tots els seus objectius estan dirigits al millor ús de l'energia solar. Com s'ha demostrat en l'estudi del cost unitari, aquest és viable econòmicament, tenint en compte que es preveu poder obtenir guanys si mai s'introdueix al mercat. Pel que fa als àmbits socials i ambientals, el regulador dissenyat aporta una millor qualitat de vida ja que preserva la vida útil de la bateria, a més a més, de mantenir un baix consum al fer-ho.

Quan es parla dels resultats, s'han obtingut els desitjats. S'espera tenir un gran impacte social, atès que la càrrega és la millor possible, i el consum de recursos que produeix és baix. Si bé la previsió de costos no ha estat tan bona com es va intuir inicialment, a causa dels obstacles que s'han anat trobant, la diferència entre el cost final i el previst és prou petita com perquè no s'hagi considerat, en cap moment, aturar el desenvolupament d'aquest.

En l'àmbit dels riscos, no es troba que n'hi hagi cap, donat que l'algorisme té una alta adaptació a tots els canvis d'escenaris possibles, sense necessitat d'aportar cap efecte negatiu a la societat ni a l'hàbitat natural. L'únic efecte negatiu que es pot pensar en la càrrega d'una bateria és la gasificació, i ja s'han pres les mesures necessàries per evitar-les.

## 9. Conclusions

Tenint en compte que aquest regulador de càrrega calcula els valors òptims de corrent en cada moment, tant per la seva ràpida càrrega com per preservar la seva vida útil el màxim temps possible, s'ha decidit no fer un estudi numèric sobre si el regulador dissenyat era més ràpid que un altre, ja que cada regulador pot fer servir un mètode diferent de càrrega.

El fet de saber que utilitza els millors valors en cada moment ja és suficient per dir que s'estan obtenint els millors resultats possibles. Un altre fet que ajuda positivament en els resultats d'aquest projecte és que, si es compara amb altres reguladors de càrrega solar, una gran majoria, en els moments més òptims de càrrega, es veuen obligats a tallar la connexió amb el panell degut a la sobretensió que produeixen aquests. Com s'ha esmentat anteriorment un panell de 12 v, per exemple, en els moments més òptims del dia pot arribar a proporcionar fins a 25V, per tant, si la bateria rebés tal voltatge es faria malbé fent imprescindible l'ús del regulador.

Gràcies a la tècnica del PWM, el regulador creat en aquest projecte s'encarrega de disminuir el voltatge rebut per la bateria, de manera que no es desaprofitin aquestes hores tan òptimes.

### 9.1. Anàlisi d'objectius

Un cop finalitzat el treball, es pot dir que s'han assolit la majoria dels objectius. Malgrat no haver dut a terme proves numèriques comparant temps de càrrega amb altres reguladors, aquest projecte és considera un èxit.

Tot i que regulador dissenyat hagi estat pensat per complir els requeriments necessaris per l'ús de bateries fins a 12V, i amb un màxim de 7,2Ah, no hi ha cap limitació per poder extrapol·lar els fets a tot tipus de bateries i panells solars.

L'únic aspecte que es necessitaria modificar per poder aplicar tot aquest coneixement a bateries de diferents voltatges seria canviar alguns dels components electrònics per d'altres més potents, per evitar malmetre el circuit.

#### 9.1.1. Objectius Materials

S'han complert tots els objectius materials d'aquest projecte ja que, com bé s'explica en l'apartat de l'algorisme, totes les funcions plantejades estan implementades. Si bé es podrien millorar algunes d'aquestes funcions, fent-les més òptimes, ràpides o nombroses, la seva realització dóna suficient confiança, per assegurar la viabilitat d'aquest projecte.

Un altre aspecte resultant molt positiu és haver obtingut un maquinari funcional de forma autònoma i de baix cost.



### 9.1.2. Objectiu Acadèmic

Els objectius acadèmics també s'han complert. L'estudiant ha desenvolupat un algorisme interactiu que proporciona el millor corrent en cada moment, sense la necessitat d'electrònica analògica, ni els seus costos addicionals de disseny, fiabilitat i circuit imprès.

Per altra banda, l'estudiant ha assolit un altre tipus de coneixements acadèmics, molt vàlids com nous conceptes d'electrònica i l'habilitat de soldar. També s'ha aprofundit molt en l'ús del microcontrolador Arduino, així com en els seus diferents tipus de connexió i ús.

### 9.2. Conclusió personal

Tot i els dubtes inicials que va tenir l'estudiant en quant a l'acceptació d'aquest projecte, degut a la manca de coneixements sobre la part electrònica que comporta, els resultats obtinguts mostren la bona decisió en acceptar-lo.

Ha estat un treball molt revelador per diferents motius. Per una banda ha servit per acabar de definir l'àmbit professional en que l'estudiant vol introduir-se. El fet d'haver estat el primer cop en desenvolupar, des de zero, un sistema encastat per ell mateix ha ajudat, en gran mesura, a aquesta decisió. Per altra banda, tota la vessant més electrònica del projecte ha permès l'exploració d'un camp nou. Per finalitzar, també ha servit per entendre la gran necessitat que hi ha en el món d'avanços en el camp de les energies renovables, ja que actualment el món es recolza massa en els combustibles fòssils.

La manca de comprovacions de temps de càrrega, fa que no es pugui saber de forma 100% fiable si aquest disseny estalvia temps o és econòmic. Malgrat això es manté una valoració molt positiva de tot el treball.

Per últim, ha estat una manera d'aprendre la importància que té una bona planificació en qualsevol projecte. La planificació inicial es basava en una idea massa optimista del seu desenvolupament, i sumat el fet que no es va mantenir un estricte seguiment d'aquesta ha comportat un endarreriment general.

### 9.3. Línies de futur

Aquest apartat és un recull d'idees per a la possible continuació del desenvolupament d'aquest regulador de càrrega.

- Afegir un mode repòs pel microcontrolador. Tot i ser un dispositiu de baix consum, com menys en tingui el regulador de la bateria més electricitat podrà ser utilitzada en altres situacions. Mantenir la pantalla LCD apagada fins que es desitgi podria ser-ne un exemple.
- Fer una anàlisi més profunda dels resultats. Tot i que es considera una idea ja viable, realitzar un estudi sobre el temps de càrrega en comparació a altres reguladors ja utilitzats pot servir millorar l'algorisme. Una possibilitat podria ser esperar més, o menys, de 5 segons abans de tornar a calcular el corrent òptim i intentar aconseguir-lo.
- Fer un sistema que controli si la cerca del millor corrent per la bateria ha entrat en un bucle o porta massa temps intentant aconseguir-lo. D'aquesta manera es podria evitar limitar el número d'intents per aconseguir el corrent desitjat.
- Afegir una alarma un cert temps abans que s'interrompi l'ús de la bateria, per tal de no perdre'l de manera tan sobtada.
- Incorporar un dispositiu sense fils per tal de poder visualitzar les dades des d'una aplicació mòbil o ordinador. També podria servir per gestionar millor el regulador, com encendre'l i apagar-lo a distància.

## 10. Glossari

Aquí es recullen alguns termes utilitzats en aquest document:

**Duty cylce:** És la relació existent entre el temps en que el senyal es troba en estat actiu i el període d'aquest.

**Corrent de flotació:** Una cop s'ha assolit el valor de tensió de plena càrrega de la bateria, el regulador injecta un corrent petit per tal de mantenir la plena càrrega. Aquest corrent injectat es denomina corrent de flotació. Així doncs aquesta corrent s'encarrega de mantenir la bateria a plena càrrega i quan no es consumeix energia té la funció de compensar la pròpia auto-descàrrega.

**Sobretensió en els panells solars:** En situacions d'orientació del sol molt directe als panells solars en hores d'alta irradiació, es pot donar el cas que la tensió dels panells estigui molt per sobre la tensió nominal.

**Sobre-consum de corrent de càrrega:** En cas que per circumstàncies de mal funcionament del sistema (avaria a la bateria, curtcircuit... ) el corrent de càrrega es situés per sobre del corrent òptim durant un temps determinat establert, desconnectarem la càrrega de la bateria.

**Baixa tensió de bateria:** La baixa tensió de bateria indica una situació de descàrrega profunda de la bateria. Aquesta comprovació es té en compte per tal d'evitar que una sobrecàrrega puntual de curta durada desactivi el consum.

**Vida útil:** És la durada, en temps, que un objecte (o dispositiu) té de vida.

**Equalització:** L'equalització reverteix els efectes de la reacció química, com l'estratificació dels electròlits, on la concentració d'àcid és més gran en el fons de la cel·la de la bateria que a la seva part superior, la qual cosa suposa una reducció de l'eficiència de la bateria.

**Gasificacions:** Reacció produïda per tensions o corrents elevats quan la bateria es troba en estat de flotació. Malmet la vida útil de la bateria en gran mesura, a més de despendre gasos tòxics.

**LCD:** *Liquid Crystal Display* (pantalla de cristall líquid).

**LED:** *Light-emitting diode* (díode emissor de llum).

**Microcontrolador:** És un petit ordinador, muntat sobre un únic circuit imprès. N'hi ha molts fabricants i models de manera que hi ha un ampli ventall de dispositius disponibles. Són totalment autònoms. Un cop programats no necessiten cap sistema informàtic i si reben l'energia elèctrica necessària funcionaran d'acord amb el seu programa intern.

**Perifèrics:** Denominació genèrica per anomenar un dispositiu o aparell auxiliar i independent connectat a la unitat central de processament, en aquest cas, el microcontrolador.

**Multímetre:** Instrument que permet mesurar diferents magnituds elèctriques, com ara la tensió (continua i alternativa) i el corrent (continu i alternatiu).

**Oscil·loscopi:** Instrument fet servir per la representació gràfica de senyals elèctrics.

**Font d'alimentació :** Dispositiu electrònic que proporciona voltatge i corrent continu per tal de poder alimentar circuits o aparells elèctrics.

**Bootloader:** És un petit programa capaç de rebre el codi d'un programa més gran i gravar-lo a la memòria del microcontrolador.

**Programador de maquinari extern:** Programa fet servir per carregar programes a microcontroladors.

## 11. Bibliografia i recursos web

- [1] Wagner, Fabian ., Pfenninger, Stefan., Gauché, Paul ., Lilliestam, Johan., Damerau, Kerstin. i Patt, Anthony. "Potential for concentrating solar power to provide baseload and dispatchable power".
- [2] SEIA®. Photovoltaic (Solar Electric). Recuperat de <http://www.seia.org/policy/solar-technology/photovoltaic-solar-electric>
- [3] Guerrero-Lemus, Ricardo. i Manuel M, José. (2012), "Concentrated Solar Power". Universidad Autónoma de Madrid. Madrid, Espanya.
- [4] REN21®. (2014). "Renewables 2014: Global Status Report". Recuperat de [http://www.ren21.net/Portals/0/documents/Resources/GSR/2014/GSR2014\\_full%20report\\_low%20res.pdf](http://www.ren21.net/Portals/0/documents/Resources/GSR/2014/GSR2014_full%20report_low%20res.pdf)
- [5] SPE®. (2015). "Global Market Outlook for Solar Power 2015 – 2019". Recuperat de <http://resources.solarbusinesshub.com/solar-industry-reports/item/global-market-outlook-for-solar-power-2015-2019>
- [6] Environment News Service (2006). "Sunny Spain to Host Europe's First Large Solar Thermal Plan".
- [7] Mattera, F., Benchetrite, D., Desmettre, D., Martin, J.L. i Potteau, E. (2003). "Journal of Power Sources". CEA Cadarache. St Paul-lez-Durance Cedex, França.
- [8] Duryea, S., Islam, S. i Lawrance, W., "A battery management system for stand alone photovoltaic energy systems". Curtin Univeristy of Technology. Perth, WA, Austràlia.
- [9] Ortega, C., González, A. i Cristín, M.A., "Sistema para monitoreo remoto y control de carga de baterías en aplicaciones fotovoltaicas", 26 Semana Nacional de Energía Solar.
- [10] Ortega, C., Orozco, J, Pacheco, J., Cristín, M.A. i Rivera A. "Monitoreo De Celdas Independientes En Arreglos Para Respaldo De Energía", Gerencia de Control e Instrumentación, Instituto de Investigaciones Eléctricas.
- [11] Rivera, A., Ortega, C., Orozco, J. i Pacheco, J. "Sistema Para Almacenar Y Analizar Los Datos Provenientes De Un Sistema De Monitoreo De Celdas Individuales En Bancos De Baterías". Gerencia de Control e Instrumentación, Instituto de Investigaciones Eléctricas.
- [12] Taheri , Nicole., Entriken , Robert. i Ye, Yinyu. "A Dynamic Linear Programming Algorithm for Facilitated Charging and Discharging of Plug-In Electric Vehicles". Stanford University. California, Estats Units.
- [13] Huang, B.J., Hsu, P.C., Wu , M.S. i Ho ,P.Y. (2010). "System dynamic model and charging control of lead-acid battery for stand-alone solar PV system". Department of Mechanical Engineering, National Taiwan University, Taipei, Taiwan.

- [14] Cristín Valdez, M.A., (2004) “Determinación del estado de carga de baterías plomo-ácido utilizando redes neuronales”. Centro Nacional de Investigación y Desarrollo Tecnológico. Cuernavaca, Mor. México.
- [15] Barr, Michael. (2001). “Pulse Width Modulation”. Embedde Systems Programming.
- [16] Morningstar Corporation®. (2000). “Why PWM?”. Recuperat de <http://support.morningstarcorp.com/wp-content/uploads/2014/07/Why-PWM-Whitepaper-January2000.pdf>
- [17] Musto, Federico. (2015, Octubre 11). “Arduino: our real boss is the community”. Universitat Jiao Tong de Shanghai.
- [18] Arduino ®. <https://www.arduino.cc/>

## 12. Annex A: Esquemes elèctrics

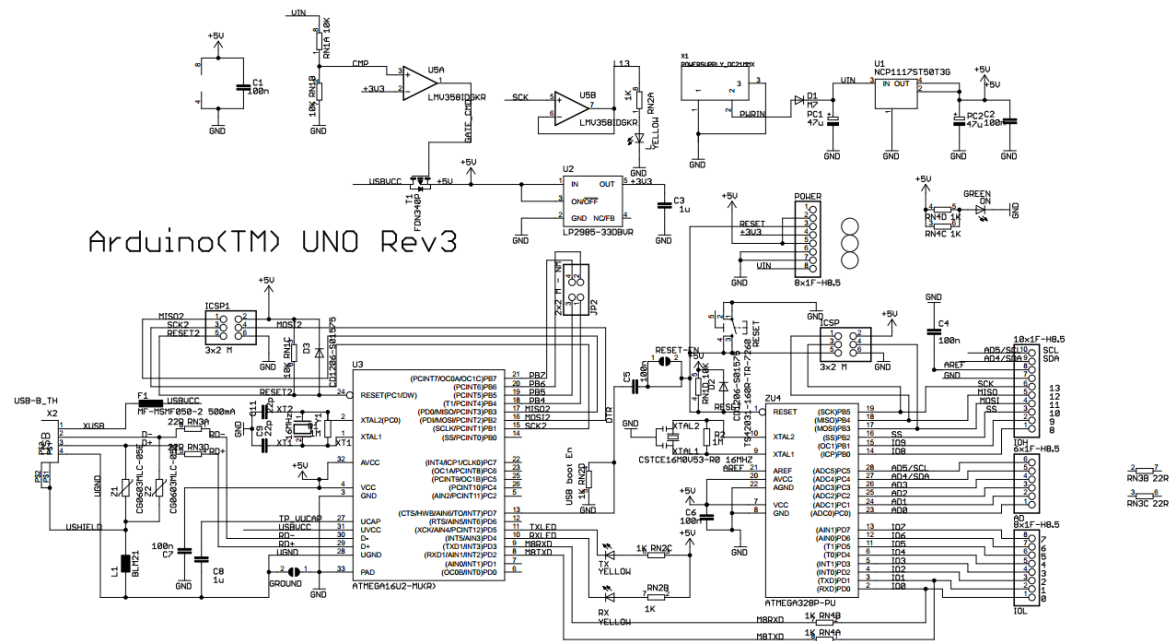


Figura 21. Diagrama Arduino Uno

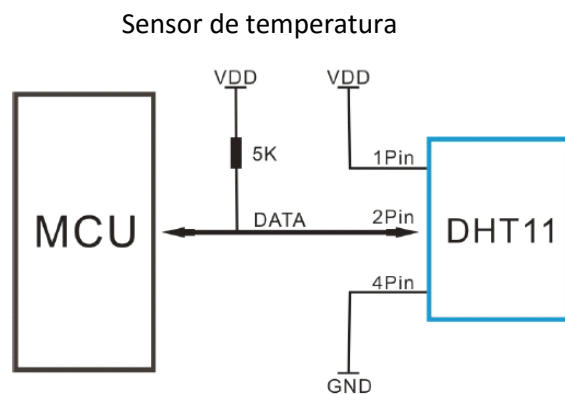


Figura 22. Diagrama sensor de temperatura DHT11

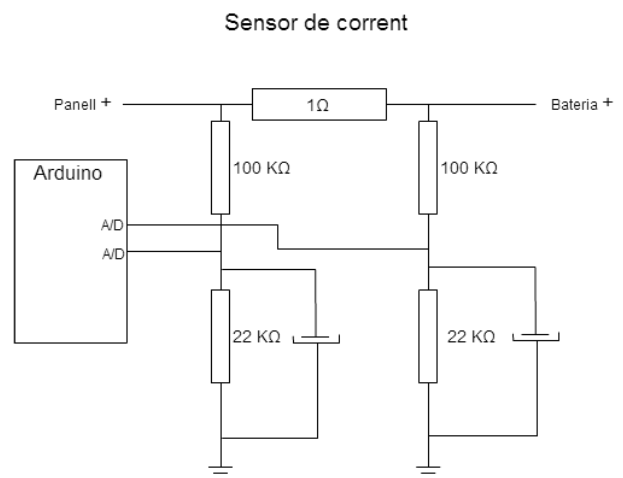


Figura 23. Diagrama sensor de corrent

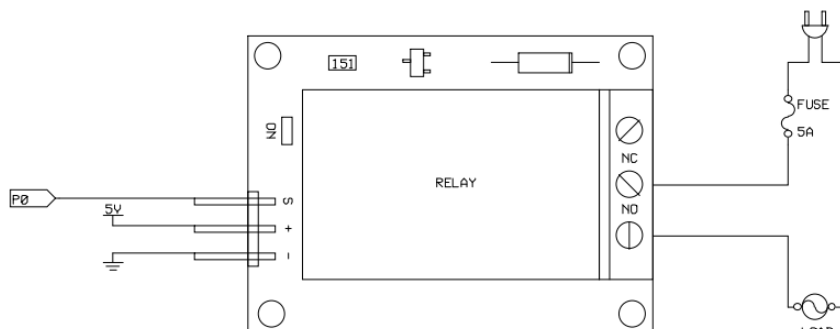


Figura 24. Diagrama relé de desconneció

### Transistor N-MOSFET

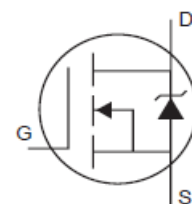


Figura 25. Diagrama transistor N-Mosfet

## 13. Annex B: Codis font

### Declaracions

```
/*----- Import needed libraries -----*/
#include "Arduino.h"
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include "DHT.h"
/*----- PINS -----*/
#define PINBUTTON1 2 //Pin for the + button
#define PINBUTTON2 3 //Pin for the - button
#define PINBUTTON3 4 //Pin for the Enter button
#define PINRELE 5 //Pin to control the relé
#define PINSPEAKER 6 //Pin for the speaker
#define PINLEDLOWBAT 8 //Pin for low battery LED
#define PINLEDHOTTEMP 9 //Pin for temp error LED
#define PINLEDCHARG 11 //Pin for charging bat LED
#define PINLEDFULLBAT 13 //Pin for bat charged LED
#define PINPWM 10 //Pin where to apply PWM
#define PINTEMPSENSOR 12 //Pin to read temp sensor
#define PINSENSORREAD1 A0 //Pin to read amps sensor 1
#define PINSENSORREAD2 A1 //Pin to read amps sensor 2
#define PINBATTERYVOLTAGE A2 //Pin to read bat voltage
#define PINPANELVOLTAGE A3 //Pin to read panel voltage

/*----- Algorithm -----*/
//--- PHASES ---
const int MINIMUM_CHARGE = 1; //phase 1
const int MAXIMUM_CHARGE = 2; //phase 2
const int OVER_CHARGE = 3; //phase 3
const int EQUALIZATION = 4; //phase 4
const int FLOATATION = 5; //phase 5

int PHASE_ONE = 35;
const int PHASE_TWO = 80;
const int PHASE_THREE = 95;
const int PHASE_FOUR = 99;
const int PHASE_FIVE = 100;

const float PHASE_ONE_PERC = 0.2;
const float PHASE_TWO_PERC = 0.4;
const float PHASE_THREE_PERC = 0.2;
const float PHASE_FOUR_PERC = 0.1;
const float PHASE_FIVE_PERC = 0.01;
//--- END PHASES ---

int currentPhase = 0;
int currentDutyCycle = 0;
const int MAXIMUM_DUTYCYCLE = 0; //0 -> Duty Cycle = 100%;
const int MINIMUM_DUTYCYCLE = 255; //255 -> Duty Cycle = 0%;

const int BATTERY_ALL_GOOD = 0;
const int HIGH_TEMP_BATTERY = 1;
const int LOW_VOLTAGE_BATTERY = 2;

int currentError = 0;
```



```
int minimumToUse = 50;
boolean toLowtoUse = false;
boolean powerToBattery = false;
boolean powerOnPanels = false;
const float R1 = 100000.0;
const float R2 = 22000.0;
const float DENOMINATOR = (R2 / (R1 + R2));
const float RESISTANCE = 1;
const float MARGIN1 = 0.1;
const float MARGIN2 = 0.05;
const float MULTIPLIER = 3;

/*----- Main program variables -----*/
float percentage;
float batteryV;
float panelV;

/*----- Battery -----*/
const double MAX_BATTERY_VOLTAGE = 12;
const double MIN_BATTERY_VOLTAGE = 6;
double maximumBatteryAmps = 7.2;
double maximumBatteryTemp = 40.0;
const int MAX_BAT_PERCENTAGE = 100;
const int MAX_REP_ALGO = 15;

const int BETWEEN_PERCENTAGE = 10;
const int STANDARD_BATTERY_V = 12; //12V as standard bat
float typeOfBattery = 2;
float batteryPercentageArray[] = {
    11.71, //0%
    11.84, //10%
    11.97, //20%
    12.10, //30%
    12.23, //40%
    12.37, //50%
    12.50, //60%
    12.63, //70%
    12.76, //80%
    12.80, //90%
    13.02 //100%
};

const int ARRAY_SIZE = (sizeof(batteryPercentageArray) / sizeof(float));

/*----- Push Button -----*/
const int BATTERY_VOLTAGE_QUESTION = 1;
const int MAXIMUM_AMPS_QUESTION = 2;
const int MINIMUM_PERCENTAGE_QUESTION = 3;
const int WAITING_PERCENTAGE_QUESTION = 4;
const int MAXIMUM_TEMP_QUESTION = 5;
int button1State = 0;
int button2State = 0;
int button3State = 0;
int lastButton1State = 0;
int lastButton2State = 0;
int lastButton3State = 0;

/*----- Temperature Sensor -----*/
#define DHTTYPE DHT11 //sensor type(DHT 11)
```

```
/*----- LCD -----*/
uint8_t batterySymbol[8] = {0x0E,0x1F,0x1F,0x1F,0x1F,0x1F,0x1F,0x1F};
uint8_t sunSymbol[8] = {0x00,0x15,0x0E,0x1F,0x0E,0x15,0x00,0x00};
uint8_t arrowSymbol[8] = {0x18,0x0C,0x06,0x03,0x03,0x06,0x0C,0x18};
const int wariningBlinks = 3;
const String emptyText = " ";

/*-----( Declare objects )-----*/
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
DHT dht(PINTEMPSENSOR, DHTTYPE);
```

## Cos principal

```
/*----- ( SETUP )-----*/
void setup()
{
  initialization_pins();
  lcd.begin(16,2);
  dht.begin();
  updatePWM(MINIMUM_DUTYCYCLE);
  updateUsageOfBattery(false);
  powerToBattery = false;
  initialization_bat();
  percentage = 0;
  batteryV = 0;
  panelV = 0;
}

/*----- ( MAIN LOOP )-----*/
void loop()
{
  panelV = readVoltagePanel();
  batteryV = readVoltageBattery();
  percentage = getbatteryPercentage(batteryV);
  applyAlgorithm(percentage);
  lcd.clear();
  lcdStatus(percentage,panelV, batteryV);
  ledStatus(percentage);
  delay(5000);
}
```

## Inicialització regulador

```

/*****
Initialization function for all the pins that will be used in the program,
to set whether they are input or output.
*****/
void initialization_pins()
{
    pinMode(PINBUTTON1, INPUT);
    pinMode(PINBUTTON2, INPUT);
    pinMode(PINBUTTON3, INPUT);
    pinMode(PINRELE, OUTPUT);
    pinMode(PINSPEAKER, OUTPUT);
    pinMode(PINLEDLOWBAT, OUTPUT);
    pinMode(PINLEDHOTTEMP, OUTPUT);
    pinMode(PINLEDCHARG, OUTPUT);
    pinMode(PINLEDFULLBAT, OUTPUT);
}

/*****
Initialization function for all the aspects that need to be set, by the
user before starting to use the solar charge controller.
*****/
void initialization_bat()
{
    voltageInitialization();
    ampsInitialization();
    lowBatteryInitialization();
    waitToUseInitialization();
    tempInitialization();
}

/*****
Display question on de LCD screen for the initialization of the battery
voltage (6V, 12V or 24V). (To be set by the user).
*****/
void voltageInitialization()
{
    lcd.createChar(3, arrowSymbol);
    boolean finish = false;
    printTextLCD(0,0,"BATTERY VOLTAGE:",0);
    printTypesBattery();
    printPointerTypesBattery();
    while(!finish)
    {
        finish = pushButton(BATTERY_VOLTAGE_QUESTION);
    }
    scaleStateOfCharge();
    lcd.clear();
}

/*****
Display question on de LCD screen for the initialization of the battery
current. (To be set by the user).
*****/
void ampsInitialization()
{
    boolean finish = false;
    printTextLCD(0,0,"MAX BAT AMPS:",0);
    printNumericLCD(0,1,maximumBatteryAmps,2,0);
}
```

```
    printTextLCD(-1, -1, "A",0);
    while(!finish)
    {
        finish = pushButton(MAXIMUM_AMPS_QUESTION);
    }
    lcd.clear();
}

/*****
Display question on de LCD screen for how low the battery percentage can
be. (To be set by the user).
*****/
void lowBatteryInitialization()
{
    boolean finish = false;
    printTextLCD(0,0,"MINIMUM BAT %:",0);
    printNumericLCD(0,1,PHASE_ONE,0,0);
    printTextLCD(-1, -1, "%",0);
    while(!finish)
    {
        finish = pushButton(MINIMUM_PERCENTAGE_QUESTION);
    }
    lcd.clear();
    if(PHASE_ONE > minimumToUse)minimumToUse = PHASE_ONE;
}

/*****
Display question on de LCD screen for how high the battery percentage has
to be before using it again, once the battery reached the low level. (To be
set by the user).
*****/
void waitToUseInitialization()
{
    boolean finish = false;
    printTextLCD(0,0,"DON'T USE UNTIL:",0);
    printNumericLCD(0,1,minimumToUse,0,0);
    printTextLCD(-1, -1, "%",0);
    while(!finish)
    {
        finish = pushButton(WAITING_PERCENTAGE_QUESTION);
    }
    lcd.clear();
}

/*****
Display question on de LCD screen for how high the battery temperature can
be. (To be set by the user).
*****/
void tempInitialization()
{
    boolean finish = false;
    printTextLCD(0,0,"MAX BAT TEMP:",0);
    printNumericLCD(0,1,maximumBatteryTemp,2,0);
    printTextLCD(-1, -1, "°C",0);
    while(!finish)
    {
        finish = pushButton(MAXIMUM_TEMP_QUESTION);
    }
    lcd.clear();
}
```

```

/*****
Controller for the change of state button detection.
*****/
boolean pushButton(int question)
{
    boolean finish = false;
    int button;
    button1State = digitalRead(PINBUTTON1);
    button2State = digitalRead(PINBUTTON2);
    button3State = digitalRead(PINBUTTON3);

    if (button1State != lastButton1State || button2State != lastButton2State
        || button3State != lastButton3State) {
        if (button1State == HIGH)
            button = PINBUTTON1;

        else if (button2State == HIGH)
            button = PINBUTTON2;

        else if (button3State == HIGH)
            button = PINBUTTON3;

        finish = changeValue(question, button);
        delay(50);
    }
    lastButton1State = button1State;
    lastButton2State = button2State;
    lastButton3State = button3State;
    return finish;
}

/*****
Change the values on the LCD screen depending on what button is pressed
and what question is being asked.
*****/
boolean changeValue (int question, int buttonPressed)
{
    boolean finish = false;
    cleanPartiallyLCD(0,1);
    switch (question)
    {
        case BATTERY_VOLTAGE_QUESTION:
            cleanPartiallyLCD(8,1);
            if(buttonPressed == PINBUTTON1)
            {
                if(typeOfBattery > (STANDARD_BATTERY_V/MAX_BATTERY_VOLTAGE))
                    typeOfBattery = typeOfBattery - 0.5;
            }
            else if(buttonPressed == PINBUTTON2)
            {
                if(typeOfBattery < (STANDARD_BATTERY_V/MIN_BATTERY_VOLTAGE))
                    typeOfBattery = typeOfBattery + 0.5;
            }
            else if(buttonPressed == PINBUTTON3) finish = true;
            printTypesBattery();
            printPointerTypesBattery();
            break;
        case MAXIMUM_AMPS_QUESTION:
            if(buttonPressed == PINBUTTON1)

```

```
        maximumBatteryAmps = maximumBatteryAmps + 0.1;
    else if(buttonPressed == PINBUTTON2)
    {
        if (maximumBatteryAmps > 0.1)
            maximumBatteryAmps = maximumBatteryAmps - 0.1;
    }
    else if(buttonPressed == PINBUTTON3) finish = true;
    printNumericLCD(0,1,maximumBatteryAmps, 2, 0);
    printTextLCD(-1,-1,"A",0);
    break;
case MINIMUM_PERCENTAGE_QUESTION:
    if(buttonPressed == PINBUTTON1)
    {
        if(PHASE_ONE < MAX_BAT_PERCENTAGE) PHASE_ONE++;
    }
    else if(buttonPressed == PINBUTTON2)
    {
        if(PHASE_ONE > 0) PHASE_ONE--;
    }
    else if(buttonPressed == PINBUTTON3) finish = true;
    printNumericLCD(0,1,PHASE_ONE, 0, 0);
    printTextLCD(-1,-1,"%",0);
    break;
case WAITING_PERCENTAGE_QUESTION:
    if(buttonPressed == PINBUTTON1)
    {
        if(minimumToUse < MAX_BAT_PERCENTAGE) minimumToUse++;
    }
    else if(buttonPressed == PINBUTTON2)
    {
        if(minimumToUse > PHASE_ONE) minimumToUse--;
    }
    else if(buttonPressed == PINBUTTON3) finish = true;
    printNumericLCD(0,1,minimumToUse, 0, 0);
    printTextLCD(-1,-1,"%",0);
    break;
case MAXIMUM_TEMP_QUESTION:
    if(buttonPressed == PINBUTTON1)
        maximumBatteryTemp = maximumBatteryTemp + 0.5;
    else if(buttonPressed == PINBUTTON2)
        maximumBatteryTemp = maximumBatteryTemp - 0.5;
    else if(buttonPressed == PINBUTTON3) finish = true;
    printNumericLCD(0,1,maximumBatteryTemp, 2, 0);
    printTextLCD(-1,-1,"*C",0);
    break;
default:
    break;
}
return finish;
}
```

## Lectures de voltatge

```
/******  
Read voltage on the pinRead (taking into account there is a voltage  
divider on the circuit.  
*****/  
float readVoltage(int pinRead)  
{  
    int valueRead = 0;  
    float vout = 0.0;  
    float vin = 0.0;  
    valueRead = analogRead(pinRead);  
    vout = valueRead * (5/1023.0);  
    vin = vout/DENOMINATOR;  
    return vin;  
}  
  
/******  
Read voltage of the battery.  
*****/  
float readVoltageBattery()  
{  
    float batteryVoltage = readVoltage(PINBATTERYVOLTAGE);  
    return batteryVoltage;  
}  
  
/******  
Read voltage of the energy source. (Solar panels)  
*****/  
float readVoltagePanel()  
{  
    float panelVoltage = readVoltage(PINPANELVOLTAGE);  
    if(panelVoltage > 0) powerOnPanels = true;  
    else powerOnPanels = false;  
    return panelVoltage;  
}
```

## Sensor de corrent

```
/******  
Read the current of the circuit (from pins PINSENSORREAD1 and  
PINSENSORREAD2).  
*****/  
float readCurrentSensor()  
{  
    float amps = 0.0;  
    float valueRead1 = readVoltage(PINSENSORREAD1);  
    float valueRead2 = readVoltage(PINSENSORREAD2);  
    amps = (valueRead1 - valueRead2) / RESISTANCE;  
    return amps;  
}
```

## Funcions de la bateria

```

/*****
 Scales state of charge for the battery, depending on what kind of battery
 will be used (Since the percentage are lineal this can be done easily.
 *****/
void scaleStateOfCharge()
{
    for(int i = 0; i < ARRAY_SIZE; ++i){
        batteryPercentageArray[i] = batteryPercentageArray[i] / typeOfBattery;
    }
}

/*****
 Returns the percentage of the battery charge, using a state of charge
 table.
 *****/
//-- 12v Battery State of charge --//
//      Voltage      (%)Charge      //
//      13.02         100            //
//      12.80         90             //
//      12.76         80             //
//      12.63         70             //
//      12.50         60             //
//      12.37         50             //
//      12.23         40             //
//      12.10         30             //
//      11.97         20             //
//      11.84         10             //
//      11.71         0              //
//----- state of charge -----*/
float getbatteryPercentage(float batteryV)
{
    float inBetween      = 0;
    float voltageDifference = 0;
    float percentage      = 0;
    boolean valueFound    = false;
    int i                 = 0;
    while (i < ARRAY_SIZE && !valueFound)
    {
        if(batteryV <= batteryPercentageArray[i]) valueFound = true;
        else ++i;
    }
    if(i > 0)
    {
        if(i >= ARRAY_SIZE) percentage = MAX_BAT_PERCENTAGE;
        else
        {
            inBetween = (batteryPercentageArray[i] - batteryPercentageArray[i-1])
/ BETWEEN_PERCENTAGE;
            voltageDifference = batteryV - batteryPercentageArray[i-1];
            if(inBetween > 0 && voltageDifference > 0)
            {
                float scaled = voltageDifference / inBetween;
                percentage = ((i - 1) * BETWEEN_PERCENTAGE) + scaled;
            }
        }
    }
    return percentage;
}

```



## Algorisme de càrrega interactiu

```
/*
Returns whether the battery is being charged or not, by checking if the
source is giving power and that we let power in into the battery.
*/
boolean isCharging()
{
    if (!powerOnPanels) return false;
    else if (!powerToBattery) return false;
    else return true;
}

/*
Updates the current phase of the charging algorithm.
*/
void updatecurrentPhase(int percentage)
{
    if (percentage <= PHASE_ONE) currentPhase = MINIMUM_CHARGE;
    else if (percentage <= PHASE_TWO) currentPhase = MAXIMUM_CHARGE;
    else if (percentage <= PHASE_THREE) currentPhase = OVER_CHARGE;
    else if (percentage <= PHASE_FOUR) currentPhase = EQUALIZATION;
    else /* (percentage == 100) */ currentPhase = FLOATATION;
}

/*
Sets the best PWM possible to charge the battery, depending on the actual
current and the optimum one.
*/
void chargeFunction(float optimumAmps)
{
    boolean optimum = false;
    float ampsCharging = 0.0;
    int cont = 0;
    if (ampsCharging >= 0)
    {
        while (!optimum && cont < MAX_REP_ALGO)
        {
            ampsCharging = readCurrentSensor();
            if (ampsCharging > optimumAmps)
            {
                if ((ampsCharging - optimumAmps) > optimumAmps * MARGIN1)
                    currentDutyCycle =
                        currentDutyCycle + ((ampsCharging / optimumAmps) * MULTIPLIER);

                else if ((ampsCharging - optimumAmps) > optimumAmps * MARGIN2)
                    currentDutyCycle++;

                else optimum = true;

                if (currentDutyCycle >= MINIMUM_DUTYCYCLE)
                {
                    currentDutyCycle = MINIMUM_DUTYCYCLE;
                    optimum = true;
                    powerToBattery = false;
                }
                else if (currentDutyCycle <= MAXIMUM_DUTYCYCLE)
                {
                    currentDutyCycle = MAXIMUM_DUTYCYCLE;
                }
            }
            cont++;
        }
    }
}
```

```
        optimum = true;
    }
}
else if (ampsCharging < optimumAmps)
{
    if ((ampsCharging - optimumAmps) < (optimumAmps*(-MARGIN1)))
        currentDutyCycle =
            currentDutyCycle - ((optimumAmps/ampsCharging) * MULTIPLIER);

    else if ((ampsCharging - optimumAmps) < (optimumAmps*(-MARGIN2)))
        currentDutyCycle--;

    else optimum = true;

    if(currentDutyCycle <= MAXIMUM_DUTYCYCLE)
    {
        currentDutyCycle = MAXIMUM_DUTYCYCLE;
        optimum = true;
    }
    else if(currentDutyCycle >= MINIMUM_DUTYCYCLE)
    {
        currentDutyCycle = MINIMUM_DUTYCYCLE;
        optimum = true;
        powerToBattery = false;
    }
}
else
{
    optimum = true;
}
updatePWM(currentDutyCycle);
delay(500);    //delay a little bit to allow the new PWM to stabilize
cont++;
}
}

/*****
 Checks if the battery percentage is too low or still not high enough to be
 used again.
 *****/
boolean checkLowBatteryVoltage(int percentage)
{
    if (percentage <= PHASE_ONE)                return true;
    else if(toLowtoUse && percentage <= minimumToUse) return true;
    else                                         return false;
}

/*****
 Check if the temperature is higher than the one set by the user.
 *****/
boolean checkHighTemperatureBattery()
{
    float temperature = checkTemperature();
    if(temperature > maximumBatteryTemp) return true;
    else                               return false;
}
```

```

/*****
Checks if there is any 'problem' with the battery.
*****/
int checkBatteryProteccion(int percentage)
{
    if(checkHighTemperatureBattery())          return HIGH_TEMP_BATTERY;
    else if(checkLowBatteryVoltage(percentage)) return LOW_VOLTAGE_BATTERY;
    else                                         return BATTERY_ALL_GOOD;
}
/*****
Applies the algorithm, by:
Calculating the optimum current, depending on the phase the algorithm is.
Checking if there is any "problem" that needs to be handled (low bat, high
temp) and handling it.
*****/
void applyAlgorithm(int percentage){
    float optimumAmps = maximumBatteryAmps * PHASE_TWO_PERC;
    updatecurrentPhase(percentage);
    switch (currentPhase){
        case MINIMUM_CHARGE:
            optimumAmps = maximumBatteryAmps * PHASE_ONE_PERC;
            break;
        case MAXIMUM_CHARGE:
            optimumAmps = maximumBatteryAmps * PHASE_TWO_PERC;
            break;
        case OVER_CHARGE:
            optimumAmps = maximumBatteryAmps * PHASE_THREE_PERC;
            break;
        case EQUALIZATION:
            optimumAmps = maximumBatteryAmps * PHASE_FOUR_PERC;
            break;
        case FLOATATION:
            optimumAmps = maximumBatteryAmps * PHASE_FIVE_PERC;
            break;
        default:
            break;
    }
    currentError = checkBatteryProteccion(percentage);
    switch (currentError){
        case BATTERY_ALL_GOOD:
            toLowtoUse = false;
            powerToBattery = true;
            updateUsageOfBattery(true);
            chargeFunction(optimumAmps);
            break;
        case HIGH_TEMP_BATTERY:
            updatePWM(MINIMUM_DUTYCYCLE);
            updateUsageOfBattery(false);
            powerToBattery = false;
            break;
        case LOW_VOLTAGE_BATTERY:
            toLowtoUse = true;
            powerToBattery = true;
            updateUsageOfBattery(false);
            chargeFunction(optimumAmps);
            break;
        default:
            break;
    }
}
}

```

## Sensor de temperatura

```
/******  
 Reads the from the temperature sensor. If the reading can't be done stop  
 the charging and display an error message with a noise.  
******/  
float checkTemperature()  
{  
    float temperature = dht.readTemperature();  
    while (isnan(temperature)) {  
        temperature = dht.readTemperature();  
        updateUsageOfBattery(false);  
        updatePWM(MINIMUM_DUTYCYCLE);  
        lcd.clear();  
        delay(500);  
        printTextLCD(0,0,"TEMPERATURE SEN.",0);  
        printTextLCD(5,1,"ERROR",0);  
        delay(500);  
        updateSound(250);  
    }  
    updateSound(0);  
    return temperature;  
}
```

## PWM

```
/******  
 Updates the PWM value  
******/  
void updatePWM(int value)  
{  
    analogWrite(PINPWM,value);  
}
```

## Relé

```
/******  
 Set the relé on or off allowing (or stoping) the battery to be used.  
******/  
void updateUsageOfBattery(boolean start)  
{  
    if(start) digitalWrite (PINRELE, HIGH);  
    else digitalWrite (PINRELE, LOW);  
}
```

## Altaveu

```
/******  
 Updates the sound the speaker is doing.  
******/  
void updateSound(int value)  
{  
    analogWrite(PINSPEAKER,value);  
}
```

## Leds

```
/******  
Shows whether the battery is charged, being charge, if the battery is too  
low or the temperature too high, using LEDS.  
*****/  
void ledStatus(int percentage)  
{  
    switch (currentError)  
    {  
        case BATTERY_ALL_GOOD:  
            if (percentage == MAX_BAT_PERCENTAGE)  
            {  
                digitalWrite(PINLEDFULLBAT, HIGH);  
                digitalWrite(PINLEDCHARG, LOW);  
            }  
            else  
            {  
                if(isCharging())  
                {  
                    digitalWrite(PINLEDFULLBAT, LOW);  
                    digitalWrite(PINLEDCHARG, HIGH);  
                }  
                else  
                {  
                    digitalWrite(PINLEDFULLBAT, LOW);  
                    digitalWrite(PINLEDCHARG, LOW);  
                }  
            }  
            digitalWrite(PINLEDLOWBAT, LOW);  
            digitalWrite(PINLEDHOTTEMP, LOW);  
            break;  
        case HIGH_TEMP_BATTERY:  
            digitalWrite(PINLEDHOTTEMP, HIGH);  
            digitalWrite(PINLEDLOWBAT, LOW);  
            digitalWrite(PINLEDCHARG, LOW);  
            digitalWrite(PINLEDFULLBAT, LOW);  
            break;  
        case LOW_VOLTAGE_BATTERY:  
            if (percentage == MAX_BAT_PERCENTAGE)  
            {  
                digitalWrite(PINLEDFULLBAT, HIGH);  
                digitalWrite(PINLEDCHARG, LOW);  
            }  
            else  
            {  
                if(isCharging())  
                {  
                    digitalWrite(PINLEDFULLBAT, LOW);  
                    digitalWrite(PINLEDCHARG, HIGH);  
                }  
                else  
                {  
                    digitalWrite(PINLEDFULLBAT, LOW);  
                    digitalWrite(PINLEDCHARG, LOW);  
                }  
            }  
            digitalWrite(PINLEDLOWBAT, HIGH);  
            digitalWrite(PINLEDHOTTEMP, LOW);  
            break;  
    }  
}
```

```
    default:
        digitalWrite(PINLEDHOTTEMP, LOW);
        digitalWrite(PINLEDLOWBAT, LOW);
        digitalWrite(PINLEDCHARG, LOW);
        digitalWrite(PINLEDFULLBAT, LOW);
        break;
    }
}
```

## LCD

```
/******
Display voltages from the battery and the source on the LCD screen.
*****/
void printVoltages(float panelV, float batteryV)
{
    printNumericLCD(1,1,batteryV, 2, 0);
    printTextLCD(-1,-1,"V",0);
    printNumericLCD(1,0,panelV, 2, 0);
    printTextLCD(-1,-1,"V",0);
}

/******
Display both status of the charging (with warning messages if needed) and
the percentage charged of the battery.
*****/
void printbatteryPercentage(float percentage)
{
    String warningText;
    if (percentage <= 9.5)
    {
        printNumericLCD(14,1,percentage, 0, 0);
        printTextLCD(-1,-1,"%",0);
    }
    else if (percentage <= 99.5)
    {
        printNumericLCD(13,1,percentage, 0, 0);
        printTextLCD(-1,-1,"%",0);
    }
    else
    {
        printNumericLCD(12,1,percentage, 0, 0);
        printTextLCD(-1,-1,"%",0);
    }

    switch (currentError)
    {
        case BATTERY_ALL_GOOD:
            if (percentage == MAX_BAT_PERCENTAGE)
                printTextLCD(9,0,"Charged",0);
            else
            {
                if (isCharging()) warningText = "Charging";
                else warningText = emptyText;
                printTextLCD(8,0,warningText,0);
            }
            break;
        case HIGH_TEMP_BATTERY:
            warningText = "HOT BAT!";
            printTextLCD(8,0,warningText,warningBlinks);
    }
}
```

```
        break;
    case LOW_VOLTAGE_BATTERY:
        if (percentage <= PHASE_ONE) warningText = "LOW BAT!";
        else warningText = "NEED BAT";
        printTextLCD(8,0,warningText,warningBlinks);
        break;
    default:
        break;
}
printTextLCD(8,0,warningText,0);
}

/*****
 Controller for the LCD displays during the charging of the battery.
 *****/
void lcdStatus(float percentage, float panelV, float batteryV)
{
    lcd.createChar(2, batterySymbol);
    lcd.createChar(1, sunSymbol);
    for (int i=0; i<2; i++)
    {
        lcd.setCursor(0,i);
        lcd.write(i+1);
    }
    printVoltages(panelV,batteryV);
    printbatteryPercentage(percentage);
}

/*****
 Prints different types of battery voltage on the displaying of the
 question about battery voltage.
 *****/
void printTypesBattery()
{
    printNumericLCD(2,1,6,0,0);
    printTextLCD(-1, -1, "V",0);
    printNumericLCD(6,1,8,0,0);
    printTextLCD(-1, -1, "V",0);
    printNumericLCD(11,1,12,0,0);
    printTextLCD(-1, -1, "V",0);
}

/*****
 Draws the pointer to choose the battery voltage on the displaying of the
 question about battery voltage. Changing the position depending on the
 current location of the pointer.
 *****/
void printPointerTypesBattery()
{
    if (typeOfBattery == 2) lcd.setCursor(1,1); //6V
    else if (typeOfBattery == 1.5) lcd.setCursor(5,1); //8V
    else if (typeOfBattery == 1) lcd.setCursor(10,1); //12V
    lcd.write(3);
}
```

```

/*****
  Displays a blank space of 8 squares on the LCD screen, starting in the
  positions (x,y) .
  *****/
void cleanPartiallyLCD(int x, int y)
{
  lcd.setCursor(x,y);
  lcd.print(emptyText);
}

/*****
  Display a number, with the decimals established, on the LCD screen in the
  position (x,y) and blinks the amount of time set by 'blinkNumber'.
  *****/
void printNumericLCD(int x, int y, float number, int decimals, int
blinkNumber)
{
  if ((x > -1) && (y > -1)) lcd.setCursor(x,y);
  lcd.print(number,decimals);
  for(int count = 0; count < blinkNumber; ++count)
  {
    cleanPartiallyLCD(x,y);
    delay(400);
    lcd.setCursor(x,y);
    lcd.print(number,decimals);
    delay(400);
  }
}

/*****
  Display a text on the LCD screen in the position (x,y) and blinks the
  amount of time set by 'blinkNumber'.
  *****/
void printTextLCD(int x, int y, String text, int blinkNumber)
{
  if ((x > -1) && (y > -1)) lcd.setCursor(x,y);
  lcd.print(text);
  for(int count = 0; count < blinkNumber; ++count)
  {
    cleanPartiallyLCD(x,y);
    delay(400);
    lcd.setCursor(x,y);
    lcd.print(text);
    delay(400);
  }
}

```